

1. CBIIT Semantic Infrastructure 2.0 Roadmap	2
1.1 1 - Invitation to Review the Roadmaps - caGrid 2.0 and Semantic Infrastructure 2.0	2
1.2 2 - Introduction to the Roadmaps - caGrid 2.0 and Semantic Infrastructure 2.0	3
1.3 3 - Stakeholders - caGrid 2.0 and Semantic Infrastructure 2.0	7
1.4 4 - Semantic Infrastructure 2.0 Use Cases	14
1.4.1 4.1 - Translational Medicine, Research and Personalized Medicine	14
1.4.2 4.2 - Life Sciences	14
1.4.3 4.3 - Clinical Trials	20
1.4.4 4.4 - Electronic Health Records	21
1.4.5 4.5 - Terminologies Use Cases	23
1.4.6 4.6 Other Use Cases	25
1.5 5 - Semantic Infrastructure Functional Requirements	27
1.5.1 5.1 Artifact Management	27
1.5.2 5.2 Service Discovery and Governance	29
1.5.3 5.3 Clinical Data Forms Definition and Modeling	30
1.5.4 5.4 Decision Support and Reasoning	31
1.5.5 5.5 Conformance Testing	31
1.5.6 5.6 caGrid 2.0 Platform and Terminology Integration	32
1.5.7 5.7 Other Functional Requirements	35
1.6 6 - Semantic Infrastructure 2.0 Architecture	37
1.6.1 6.1 - Overview of Semantic Infrastructure 2.0 Architecture	38
1.6.2 6.2 - Overview of Semantic Infrastructure 2.0 Capabilities and Services	40
1.6.3 6.3 - Tools for Semantic Infrastructure 2.0	42
1.6.4 6.4 - Tie-in with Terminology and Platform	44
1.7 7 - Gap Assessment for Semantic Infrastructure	46
1.8 8 - Migration Strategy and Ongoing Support for Existing Customers	48
1.9 9 - CBIIT Project Recommendations	48
1.10 10 - Semantic Infrastructure 2.0 Interim Development	51
1.11 11. Semantic Infrastructure 2.0 Roadmap References and Glossary	51

CBIIT Semantic Infrastructure 2.0 Roadmap

CBIIT Semantic Infrastructure 2.0 Roadmap

February 23, 2011 Working Draft

The Semantic Infrastructure 2.0 Roadmap includes the following:

- 1 - Invitation to Review the Roadmaps - caGrid 2.0 and Semantic Infrastructure 2.0
- 2 - Introduction to the Roadmaps - caGrid 2.0 and Semantic Infrastructure 2.0
- 3 - Stakeholders - caGrid 2.0 and Semantic Infrastructure 2.0
- 4 - Semantic Infrastructure 2.0 Use Cases
 - 4.1 - Translational Medicine, Research and Personalized Medicine
 - 4.2 - Life Sciences
 - 4.3 - Clinical Trials
 - 4.4 - Electronic Health Records
 - 4.5 - Terminologies Use Cases
 - 4.6 Other Use Cases
- 5 - Semantic Infrastructure Functional Requirements
 - 5.1 Artifact Management
 - 5.2 Service Discovery and Governance
 - 5.3 Clinical Data Forms Definition and Modeling
 - 5.4 Decision Support and Reasoning
 - 5.5 Conformance Testing
 - 5.6 caGrid 2.0 Platform and Terminology Integration
 - 5.7 Other Functional Requirements
- 6 - Semantic Infrastructure 2.0 Architecture
 - 6.1 - Overview of Semantic Infrastructure 2.0 Architecture
 - 6.2 - Overview of Semantic Infrastructure 2.0 Capabilities and Services
 - 6.3 - Tools for Semantic Infrastructure 2.0
 - 6.4 - Tie-in with Terminology and Platform
- 7 - Gap Assessment for Semantic Infrastructure
- 8 - Migration Strategy and Ongoing Support for Existing Customers
- 9 - CBIIT Project Recommendations
- 10 - Semantic Infrastructure 2.0 Interim Development
- 11. Semantic Infrastructure 2.0 Roadmap References and Glossary

1 - Invitation to Review the Roadmaps - caGrid 2.0 and Semantic Infrastructure 2.0

1 - Invitation to Review the Roadmaps - caGrid 2.0 and Semantic Infrastructure 2.0

February 23, 2011 Working Draft

This page invites you to review the roadmap and explains the importance of your review. The team also reports on development to date and plans for continued development.

- [Invitation to Review the Semantic Infrastructure 2.0 Roadmap](#)
- [Planned Development of the Semantic Infrastructure 2.0 Roadmap](#)

Invitation to Review the Semantic Infrastructure 2.0 Roadmap

In order to capitalize on new informatics technologies and changing community needs, we are asking for collaboration in the development of two roadmaps:

- [caGrid 2.0 Roadmap](#)
- [Semantic Infrastructure 2.0 Roadmap](#)

As a core stakeholder in caBIG® tools and technologies, your input is essential in guiding the development of expanded capabilities to support the rapidly evolving needs of the caBIG® community. Please review this draft and provide your comments on the Community Input form spreadsheet, available for download for the [caGrid 2.0 Roadmap](#) and the [Semantic Infrastructure 2.0 Roadmap](#).

Building on a combination of current caGrid and caBIG® semantic infrastructure technology, advances in technology that have occurred over the past several years, and lessons learned from our collective experiences with caGrid 1.x and with the current Semantic Infrastructure including caDSR and EVS, the vision for caGrid 2.0 and Semantic Infrastructure 2.0 is to provide enhanced capabilities in the context of three overarching requirements:

- Lower the current barrier-to-entry to use of the caBIG® tools and technologies
- Provide a "linear value proposition" to caBIG® stakeholders – "make easy things easy to do"
- Provide support for users of the first-generation caBIG® infrastructure and their data

Please review this working draft and give us your input regarding the scope for this roadmap, specific capabilities that you see as essential for success, or suggestions for improvement of the document. The PST project team will review and post the disposition of all content-related input received for the caGrid 2.0 Roadmap and the Semantic Infrastructure 2.0 project team will do the same for the Semantic Infrastructure 2.0 Roadmap.

Each team will publish ongoing evolutions of the document to reflect substantive content changes as a result of both team and community input. Each version will be publicly available for comment.

Thank you in advance for your contributions and ongoing engagement as we collaboratively develop the capabilities that will define caGrid 2.0 and the Semantic Infrastructure 2.0. This opportunity to collaborate and share your expertise will ensure that we can deliver the best possible product to the broad spectrum of users in the caBIG® community.

caGrid 2.0 Roadmap Team

Charlie Mead
Platform/Security/Tooling Project Lead
meadch@mail.nih.gov

Robert Shirley
Platform/Security/Tooling Sponsor
robert.shirley@nih.gov

Semantic Infrastructure 2.0 Roadmap Team

Charlie Mead
Semantic Infrastructure 2.0
Project Lead
charlie.mead@nih.gov

Dave Hau
Semantic Infrastructure 2.0
Project Sponsor
dave.hau@nih.gov

Planned Development of the Semantic Infrastructure 2.0 Roadmap

Teams have been working on developing the caGrid 2.0 and Semantic Infrastructure 2.0 roadmaps for several weeks, and drafts have been published with requests for input from the community. The teams appreciate the input received and want to report on evolution of development.

Initially, one team was developing a roadmap for caGrid 2.0 and the other a roadmap for the Semantic Infrastructure 2.0. The purpose was conceived as defining the high-level requirements and prototypical architecture for the next generation of the caBIG® infrastructure.

As the two Roadmap projects have matured, it has become clear that the most effective use of the expertise represented on the two roadmap project teams is to charge the teams with developing all of the documentation that would normally be produced by a somewhat more robust software development lifecycle Inception Phase. Thus the project is embarking on the formal start of an extended software engineering process. This includes high-level requirements, formally-defined scope and vision, and a substantial effort directed at risk identification, profiling, and reduction and mitigation of risk, including development of prototypes when necessary to more completely understand a given risk or collection of risks.

Therefore the planned termination date for the two "roadmap" projects is being extended to more closely coincide with the issuing of the Requests for Proposals and associated Statements of Work. This will enable responders to use the output of the Inception Phase roadmap teams as input when they plan, and ultimately execute, contracted Elaboration, Construction, and Transition phases for the various caGrid 2.0 and Semantic Infrastructure 2.0 components described in the Inception Phase documentation.

Specifically, the Inception Phase effort will include continued refinement of the existing roadmap documents, additional requirements gathering and traceability mapping, risk assessment and mitigation, and scope definition activities *in parallel* with the development of the Request for Proposals (RFPs).

2 - Introduction to the Roadmaps - caGrid 2.0 and Semantic

Infrastructure 2.0

2 - Introduction to the Roadmaps - caGrid 2.0 and Semantic Infrastructure 2.0

February 23, 2011 Working Draft



Note on the Scope of the Roadmaps: What they are and what they're not

The two roadmaps should be viewed in the context of the larger software development lifecycle that will produce the components of caGrid 2.0 infrastructure and Semantic Infrastructure 2.0 infrastructure and tools. The roadmaps are **inception phase** documents. The primary purpose of the roadmaps is to **define the Scope and Vision** for the development lifecycle. This includes **identification of relevant stakeholders**, and **high-level "business use cases."** In addition, inception phase artifacts should focus on identifying and whenever possible mitigating, significant project **risks**. The two roadmaps are *not requirements documents*, artifacts that will be developed following the release of and contracting for one or more Requests for Proposals (RFPs). The scope and trajectory of the RFPs are being informed by the roadmaps. To attempt to make the roadmaps into detailed requirements documents is to miss the importance of inception phase scope and vision, stakeholder, and risk documentation – the focus of the caGrid 2.0 and Semantic Infrastructure 2.0 Roadmaps.

This introduction addresses *Next-Generation Goals and Objectives for caGrid 2.0 and Semantic Infrastructure 2.0* and includes the following:

- Overarching Central Requirements
 - Lower the current barrier-to-entry to use of the caBIG® tools and technologies
 - Provide a "linear value proposition" to caBIG® stakeholders - "make easy things easy to do"
 - Provide support for users of the first-generation caBIG® infrastructure and their data
- Defining the Scope: "Capability Big Buckets" - Use Cases, Story Boards, High-Level Business Requirements
 - Security
 - Discovery (including *ad hoc* and distributed queries)
 - Composability
- Architecture Paradigm: Semantically-Aware Service-Oriented Architecture (sSOA)
- Service-Aware Interoperability Framework (SAIF)
 - Scalable Performance
 - Standards-awareness
- Additional Information about Use of sSOA and SAIF

Overarching Central Requirements

The caBIG® Next-Generation Grid and Semantic Infrastructure (caGrid 2.0 and Semantic Infrastructure 2.0) are being developed to provide the necessary technical and semantic infrastructure to support the evolving needs of the caBIG® community – scientists, clinicians, trialists, patients, and other caBIG® stakeholders.

The caGrid 2.0 and Semantic Infrastructure 2.0 roadmap projects are focused on satisfying three overarching requirements that are central to success.

- Lower the current barrier-to-entry to use of the caBIG® tools and technologies
- Provide a "linear value proposition" to caBIG® stakeholders – "make easy things easy to do"
- Provide support for users of the first-generation caBIG® infrastructure and their data

The following provides information about these overarching requirements.

Lower the current barrier-to-entry to use of the caBIG® tools and technologies

The goal is for all stakeholders to be able to use the next generation of both caGrid and its closely-associated Semantic Infrastructure more efficiently and effectively to accomplish the tasks they want and need to accomplish.

Provide a "linear value proposition" to caBIG® stakeholders – "make easy things easy to do"

From an architecture and infrastructure perspective, this requirement is related to – but separate from – lowering the barrier-to-entry. Implicit in making "easy things easy to do" are several inter-related concepts. For caGrid 2.0 these include:

- Support for multiple levels and layers of interoperability and security ("just enough security")
- Support for more than one programming model (for example, one that utilizes a SOAP messaging protocol or one that follows a REST design pattern) for development of, and communication between, caGrid 2.0 services
- Support for enhanced workflow definition and execution capabilities including runtime service functional behavior based on design-time behavioral semantics
- Layered, runtime integration of the caGrid 2.0 "tech stack" with the Semantic Infrastructure 2.0 "semantic stack"

Layered integration allows caBIG® stakeholders to deploy and use caBIG® tools and technologies with "just enough security, semantics, and

specifications" to accomplish the desired task in the particular context.

For the Semantic Infrastructure 2.0, the central concepts and requirements include:

- Support for multiple levels and layers of semantic robustness ("just enough semantics")
- Support for "flexible enhanced relationship management," that is, ability to define at an appropriate level of granularity the semantics of concept-to-concept relations to allow for computational representation of context
- Support for layered semantic representations to facilitate extensible reasoning
- Support for *ad hoc* queries

The combination of (1) the experience gained with the first-generation of caBIG® infrastructure – caGrid 1.x, caDSR, and EVS – and (2) the substantial advances that have been made in a number of related technologies in the time since the various components of the first-generation of caBIG® infrastructure, both frames and grounds the feasibility and reality of satisfying these two overarching, central requirements.

Provide support for users of the first-generation caBIG® infrastructure and their data

This third requirement is of equal importance and underscores the evolutionary nature of caGrid 2.0 and the Semantic Infrastructure 2.0. These are steps from the first-generation of caBIG®, and *not* a wholesale discarding of the past work and experience.

This requirement can be met in a number of ways ranging from "invisible-to-the-user" infrastructure management via process facades or semantically-equivalent representations of legacy data, to process migration, data migration, or both from current 1.x to 2.0 implementations.

Defining the Scope: "Capability Big Buckets" - Use Cases, Story Boards, High-Level Business Requirements



Note

Use of the term "requirements" in this section is *not* meant to be synonymous with "software system functional or non-functional and quality requirements," as these requirements are of a different type from those documented in the roadmaps. The software requirements will be found in other artifacts outside the scope of the roadmaps. The "requirements" found in the roadmaps are statements of *coarsely-granulated descriptions of what users need to be able to accomplish--customer requirements rather than software requirements, derived from the use cases, or scenarios of customers' work*. Detailed functional requirements and non-functional and quality requirements essential to software architecture will, of course, ultimately be traceable to the "requirements" in the roadmaps. Their specific statements, however -- such as "the system shall," "the system will," and "the Fit Metric is," will emerge later in the software development lifecycle. These specific requirements statements will be developed under contracts awarded following the issuing of Requests for Proposals which were informed by the roadmaps, with respect to scope, general project trajectory, and risk.

For both the caGrid 2.0 and Semantic Infrastructure 2.0 projects, a number of "capability big buckets" have emerged to address the overarching requirements.

Each "bucket" is a somewhat loosely defined logical set of requirements which will ultimately be manifest in a number of caGrid 2.0 and Semantic Infrastructure 2.0 services, or in other software functionality provided by the "tech stack" or the integrated "semantics stack."

The requirements sorted into the various "capability big buckets" have been drawn from enterprise-relevant use cases and storyboards and both roadmap documents provide traceability to explicitly document this critically important link. Traceability to requirements assures that the next-generation caBIG® infrastructure will be grounded in the requirements of the caBIG® and Translational Medicine community.

These "buckets" are listed in the sections that follow as an open invitation for discussion across the caBIG® community of their relevance and for more detailed composition through the community input process for developing the roadmaps:

- [caGrid 2.0 Roadmap](#)
- [Semantic Infrastructure 2.0 Roadmap](#)

Through this discussion, these "buckets" may be elaborated, constrained, or if necessary, deleted.

Security

Security capabilities required include the following:

- Service-level, as opposed to application-level, security capabilities
- Support for federated, policy-compliant, and scalable security capabilities
- Integrated security functionality
 - This encapsulates most of security-based functionality. Thus developers can assume that an "acceptable" level of "base-line" security can be provided by a given caGrid 2.0 service.
 - Note that the notion of "acceptable level of base-line security" may itself be developer-defined within the larger construct of the requirement that caGrid 2.0 provide a "linear value proposition." Service developers will be able to easily define the level of security necessary for a given service in a particular deployment context, ideally at both designtime and runtime.
- Ease-of-access to various service-level security parameters for manipulation by authorized individuals without underlying technical knowledge, ideally at both designtime and runtime.

Discovery (including *ad hoc* and distributed queries)

Discovery refers to locating appropriate design-time and runtime data and metadata. Such information will enable human and computationally-mediated discovery of caGrid 2.0 resources, including both data and functionality.

Discovery capabilities must be integrated within larger enterprise knowledge management constructs. Many of these constructs are described in more detail in the Semantic Infrastructure 2.0, such as the knowledge repository and its associated capabilities. Thus the notion of Discovery includes but is not necessarily limited to data elements, case report forms (CRFs), and service specifications.

It is anticipated that critical aspects of the collective set of discovery and query requirements will be enabled by the adoption and adaptation of a number of Semantic Web technologies including Resource Description Framework (RDF), Web Ontology Language (OWL), and SPARQL Query Language (SPARQL).

Composability

Both caGrid 2.0 and Semantic Infrastructure 2.0 services will be developed and deployed so as to be available for composition by non-technologists to support the development of user-defined workflows. Such workflows support scientist, clinical, and other translational medicine goals.



Note

Service orientation principles are important and should be applied at all levels in the Service Delivery Life Cycle. Of that, there is no doubt. However, CBIIT is *currently* scoping the application of SOA and its associated design principles to "enterprise services". The principles will be applied to development of APIs and tools, scoped to the overarching "enterprise services" context of the CBIIT sSOA (which will, of course, involve a number of service APIs and associated tools).

The notion of "applications as compositions of services focused on accomplishing a particular business requirement" is one of the central constructs of the overarching semantically-aware Service-Oriented Architecture (sSOA) approach that CBIIT is adopting for caGrid 2.0 and Semantic Infrastructure 2.0. This approach – outlined in a bit more detail below – is grounded in industry experience that clearly demonstrates that the Semantically-Aware Service-Oriented Architecture (sSOA) design paradigm effectively supports ongoing interoperability in the context of the dynamic change and evolution that characterizes the caBIG® community.

Architecture Paradigm: Semantically-Aware Service-Oriented Architecture (sSOA)

Both the Semantic Infrastructure 2.0 and caGrid 2.0 will be developed and deployed within the context of an overarching approach to enterprise architecture which uses the distributed computing design paradigm commonly referred to as Service-Oriented Architecture (SOA).

In addition the SOA being developed by NCI CBIIT, manifested in the Semantic Infrastructure 2.0 and caGrid 2.0, is referred to as a "semantically-aware SOA" (sSOA). This addresses the fundamental importance of semantics in any architecture in the broadest possible context of the life sciences and healthcare.



Note about SOA

It is beyond the scope of this document to discuss in detail the various benefits and goals, central organizing motivations, or fundamental design principles of SOA. However, the following bullet points summarize each of these topics. Interested readers can refer to a number of references including two texts by Thomas Erl: *Principles of Service Design* and *SOA Design Patterns*."

The organizing principles of SOA are:

- Business-driven
- Vendor-neutral
- Enterprise-centric
- Composition-centric

The benefits and goals of SOA are:

- Intrinsic interoperability
- Increased federation
- Increased business and technology alignment
- Increased vendor-diversification options
- Increased IT ROI
- Decreased IT burden
- Increased organizational agility

The SOA design principles are:

- Standard Service Contracts
- Service Loose Coupling
- Service Abstraction
- Service Reusability

- Service Autonomy
- Service Statelessness
- Service Discoverability
- Service Composability

Service-Aware Interoperability Framework (SAIF)

The Semantic Infrastructure 2.0 is in large part the operational support for the metadata defined in the [CBIIT HL7 Service-Aware Interoperability Framework Implementation Guide \(SAIF IG\)](#).

Readers interested in the specifics of the metadata defined by the CBIIT SAIF Implementation Guide should consult that [document](#). In particular, the chapter on the Enterprise Conformance and Compliance Framework (ECCF) provides a focal point for the definition and representation of the collective set of informational and behavioral metadata which the Semantic Infrastructure 2.0 will support at both designtime and runtime via the caGrid 2.0 platform.

Key characteristics of the CBIIT sSOA are described in the following sections.

Scalable Performance

This includes the ability to support high-volume data processing, at the terabyte to petabyte scale.

Standards-awareness

Appropriate informatics and technical standards will be applied to enable broad-based intra-community and inter-enterprise interoperability.

Note that the "application of appropriate standards" ideally should be a decision that can be made based on the "just enough" context-sensitive criteria mentioned above with respect to "just enough security, semantics, and specifications." Therefore application of standards should be amenable to adjustment and modulation as the community-of-interest for a given caGrid 2.0 service evolves.

Additional Information about Use of sSOA and SAIF

Visit [caGrid 2.0 and Semantic Infrastructure 2.0 - Reference Frameworks for Development](#) for additional information.

3 - Stakeholders - caGrid 2.0 and Semantic Infrastructure 2.0

3 - Stakeholders - caGrid 2.0 and Semantic Infrastructure 2.0

February 23, 2011 Working Draft

This page includes the following:

- [Development of the Stakeholders List](#)
- [Definitions for the Stakeholders List](#)
- [List of Stakeholders](#)
- Stakeholder Profiles: Goals, Concerns and Expectations
 - [Executive Decision Makers](#)
 - [Resource Providers](#)
 - [caGrid Developers](#)
 - [Semantic Infrastructure Developers](#)
 - [Semantic Curators](#)
 - [Service Developers](#)
 - [Service Orchestration Developers](#)
 - [High Performance Computing \(HPC\) Pipeline Creators](#)
 - [Informaticians](#)
 - [Bench Scientists](#)
 - [Collaborators](#)
 - [Patients](#)
 - [Patient Advocates](#)

Development of the Stakeholders List

A federated, semantically aware Service-Oriented Architecture (sSOA) enables development of a continuous and accelerated cycle of scientific discovery, diagnostics, pharmaceutical product development and improved clinical care. This supports diverse stakeholders, from patients to providers, researchers to regulators, administrators to advocates, and enables effective collaborations among them.

caGrid 2.0 is an underlying reference IT connectivity platform that implements sSOA. Semantic Infrastructure 2.0 provides the underlying

semantics. caGrid 2.0 and Semantic Infrastructure 2.0 must address the entire range of stakeholders who will be participating in development and utilization of the CBIIT sSOA implementation.

The caGrid 2.0 and Semantic Infrastructure 2.0 stakeholder list in the sections that follow defines the initial list of stakeholders critical to success. The stakeholders list is being enhanced during the community review of the caGrid 2.0 and Semantic Infrastructure 2.0 roadmaps, to ensure that the list includes the full range of community members that will be affected by the roadmaps.

Definitions for the Stakeholders List

This list identifies stakeholder roles based on stakeholder category, groups within each category, and further sub-groups based on distinct roles. Note that individual members of the community may fill multiple roles simultaneously.

Following are definitions of columns in the table:

Stakeholders: This defines the high level role of the stakeholder. The following stakeholder categories have been identified:

- **Executive Decision Makers:** This category consists of stakeholders who are primary decision makers in organizations seeking to promote, adopt or adapt caGrid 2.0, Semantic Infrastructure 2.0, or both in their organizations.
- **Resource Providers:** This category consists of stakeholders who will be required to commit resources towards building caGrid 2.0, Semantic Infrastructure 2.0, or both.
- **caGrid Developers:** This category consists of stakeholders who develop the caGrid 2.0 platform under contract from CBIIT.
- **Semantic Infrastructure Developers:** This category consists of stakeholders who develop Semantic Infrastructure tools.
- **Curators:** This category consists of stakeholders who develop and refine metadata.
- **Solution Developers:** This category consists of stakeholders who will directly consume the artifacts developed for caGrid 2.0, Semantic Infrastructure 2.0, or both to create solutions for end users. Solution Developers is a broad category including, but not limited to, Service Developers and Application Developers. Groups in this category are listed in the table below. Since caGrid 2.0 is the underlying information technology (IT) platform, it is likely that these stakeholders will be mainly (though not exclusively) hands-on practical users of caGrid 2.0 technology.
- **End Users:** This category consists of stakeholders who will consume the artifacts created by the Solution Developers to achieve an end result in translational medicine. Potential end users include a broad range of specialists such as oncologists, pathologists, informaticians, citizen researchers, and IT staff responsible for deployment and security.
- **Collaborators:** This category consists of parties who are engaged in similar activity, where collaboration will help caGrid 2.0, Semantic Infrastructure 2.0, or both in reducing effort, or providing an enhanced solution.
- **Value Added Providers:** This category consists of health care IT vendors and others who will be consuming the open specifications developed by the caGrid 2.0 team, Semantic Infrastructure 2.0 team, or both teams, and will provide their own interoperable implementations.
- **Patients:** This category consists of people who receive care.
- **Patient Advocates:** This category consists of people who provide support in some way on behalf of patients.
- **Regulators:** This category consists of stakeholders who create, implement, or both, policy guidelines that impact caGrid 2.0, Semantic Infrastructure 2.0, or both directly or indirectly.

Groups: This defines different groups of stakeholders within a category of stakeholders, based on their role in the entire health care enterprise. Refer to the table for the current list of groups.

Sub-Groups: Each group could be further subdivided into smaller sub-groups, that would help determine an appropriate focused and targeted engagement plan, and ensure that the stakeholder requirements are considered at the tactical level. Refer to the table for the current list of sub-groups.

List of Stakeholders

Stakeholders	Groups	Sub-Groups
Executive Decision Makers	NCI	CBIIT Director
Executive Decision Makers	NCI	Other NCI Divisions, Programs or Projects
Executive Decision Makers	Cancer Center/Community Center (NCCCP)	Early implementer site, seeking to adopt solution
Executive Decision Makers	Cancer Center/Community Center (NCCCP)	Early implementer site, seeking to adapt solution into their environment
Executive Decision Makers	Cancer Center/Community Center (NCCCP)	Non-Early Adopter
Executive Decision Makers	Pharma and BioPharma	
Resource Providers	CBIIT Director	

Resource Providers	Value Added Providers	Vendors or sites which implement some or all components of caGrid 2.0 will need to commit resources
caGrid Developers	HealthCare IT Vendor(s)	
Semantic Infrastructure Developers	HealthCare IT Vendor(s)	
Semantic Curators	Metadata, terminology and forms curators	
Solution Developers	Service Developers	CBIIT sSOA developers
Solution Developers	Service Developers	Value Added Providers. Non funded developers (for example, open-source, academic)
Solution Developers	Service Orchestration Developers	
Solution Developers	High Performance Computing Pipeline Creators	
Solution Developers	Application Developers	CBIIT Application developers
Solution Developers	Application Developers	Value Added Providers. Non funded developers (for example, open-source, academic)
Solution Developers	Service Deployers	
Solution Developers	Application Deployers	
Solution Developers	Portal Developers	
Solution Developers	Service Conformance Verifiers	
Solution Developers	Application Administrators	
Solution Developers	Site Administrators	
Solution Developers	Site Policy Administrators	
Solution Developers	Analysts	Business Process Analysts, Service-oriented Architecture SOA Analysts, Systems Analysts
End Users	Clinicians	Oncologists
End Users	Clinicians	Pathologists
End Users	Clinicians	Radiologists
End Users	Clinicians	Disease or tumor registry SME
End Users	Clinicians	Other Oncology related professional groups
End Users	Clinical Researchers and Trialists	
End Users	Bench Scientists	
End Users	Informaticians	Science PI and Staff
End Users	Citizen Researchers	
End Users	IT Staff	Business Analyst, Deployment Staff

End Users	System Administrators and Trainers	Infrastructure Staff
End Users	Information Systems Security Office (ISSO)	
Collaborators	Standards development organizations	Health Level Seven (HL7), Clinical Data Interchange Standards Consortium (CDISC), Biomedical Research Integrated Domain Group (BRIDG), Object Management Group (OMG)
Collaborators	Standards development organizations	Federal standards bodies related to the overall Office of the National Coordinator (ONC) framework
Collaborators	Standards development organizations	International Standards Organization (ISO) , Digital Imaging and Communications in Medicine (DICOM), World Wide Web Consortium (W3C), Open Health Tools (OHT), Other
Value Added Providers		
Patients		
Patient Advocates		
Regulators	Food and Drug Administration (FDA)	
Regulators	National Institute of Standards and Technology (NIST)	
Regulators	Office of Management and Budget (OMB)	Federal Information Security Management Act (FISMA), Capital Planning and Investment Control (CPIC)
Regulators	Office of the National Coordinator (ONC)	Nationwide Health Information Network (NHIN)
Regulators	Privacy Officers	Data Sharing Group

Stakeholder Profiles: Goals, Concerns and Expectations

Representative members of several stakeholder categories were asked to list their top concerns and expectations for caGrid 2.0 and Semantic Infrastructure 2.0.

The team welcomes additional comments from members of each category. Information is added continuously.

Executive Decision Makers

Stakeholder	Executive Decision Makers
Stakeholder Goals	To align business with IT. To increase organizational agility.
Top Concerns	<ul style="list-style-type: none"> • Cost • Usability • Privacy • Content (value of) • Enterprise Class Maturity • Premature Obsolescence • Correct (sellable) Value Proposition
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Ease of adoption • Seamless use of next generation sequence information • Relating and processing diverse datasets (molecular and clinical) in a robust scalable way

Resource Providers

Stakeholder	Resource Providers
Stakeholder Goals	To maximize ROI.

Top Concerns	<ul style="list-style-type: none"> • Cost • ROI
Top Needs from caGrid2.0	<ul style="list-style-type: none"> • Facilitate more publications, citations

caGrid Developers

Stakeholder	caGrid Developers
Stakeholder Goals	To develop functional robust infrastructure.
Top Concerns	<ul style="list-style-type: none"> • Flexibility and adaptability to changing needs • Stability of platform • Are there real business use cases? • Lack of adoption • Balancing conflicting goals (for example, scope versus simplicity)
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Adoption is high • Business and scientific use case that tips the scale (has significant impact)

Semantic Infrastructure Developers

Stakeholder	Semantic Infrastructure Developers
Stakeholder Goals	To ...
Top Concerns	<ul style="list-style-type: none"> • One • Two • Three • Four • Five
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • One • Two
Top Needs from Semantic Infrastructure 2.0	<ul style="list-style-type: none"> • One • Two

Semantic Curators

Stakeholder	Semantic Infrastructure Developers
Stakeholder Goals	To ...
Top Concerns	<ul style="list-style-type: none"> • One • Two • Three • Four • Five
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • One • Two

Top Needs from Semantic Infrastructure 2.0	<ul style="list-style-type: none"> • One • Two
---	--

Service Developers

Stakeholder	Service Developers
Stakeholder Goals	To develop services that meet end users functional requirements.
Top Concerns	<ul style="list-style-type: none"> • Familiar development model • Quality factors • Simplicity and ease of use • Specification stability • Stability of basic grid functionality
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Focus on business logic

Service Orchestration Developers

Stakeholder	Service Orchestration Developer
Stakeholder Goals	To create aggregations of services which perform useful workflows.
Top Concerns	<ul style="list-style-type: none"> • Ease of discovery of capabilities and services • Complexity of data transformations • Lack of useful services • Service non-availability
Top Needs from caGrid2.0	<ul style="list-style-type: none"> • Discovery of existing orchestrations • Assurance that orchestration will work at run-time and feedback on why not (at design time) • Orchestration needs to be usable by knowledgeable subject matter expert

High Performance Computing (HPC) Pipeline Creators

Stakeholder	HPC Pipeline Creators
Stakeholder Goals	To develop and provide computational tools and services that solve computation-intensive tasks.
Top Concerns	<ul style="list-style-type: none"> • Knowledge about appropriateness of a particular tool or approach for a task • Consistency and compatibility of data access across HPC and SOA
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Choice to either bring computation to data or data to computation depending upon problem

Informaticians

Stakeholder Category	Informaticians
Stakeholder Goals	To apply computational techniques to further biomedical research.
Top Concerns	<ul style="list-style-type: none"> • Does this have the functionality I need? • Ease of use • Can I modify it easily to fit my specific needs?

Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Supply the functionality I need • Easy discovery of the functionality I need • Easily adaptable and configurable
----------------------------------	--

Bench Scientists

Stakeholder Category	Bench Scientists
Stakeholder Goals	To conduct experiments and capture data for further analysis in support of medical research.
Top Concerns	<ul style="list-style-type: none"> • How does this help me analyze my data? • Ease of use • Handle high throughput data
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Let me easily find the data and functionality that I need.

Collaborators

Stakeholder Category	Collaborators
Stakeholder Goals	To develop standards and techniques in support of scientific research.
Top Concerns	<ul style="list-style-type: none"> • Inability to perform cross-database searches
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Support for cross-database searches • Global mapping of resources and available information and data • Validated system for easily mapping among ontologies for informatics • A system to validate user privileges to eliminate the need to independently log in to each database

Patients

Stakeholder Category	Patients
Stakeholder Goals	To understand their condition, options for treatment and possible side effects.
Top Concerns	<ul style="list-style-type: none"> • Is anyone doing a trial in which I might want to participate? • Will the treatment I'm getting work for me? • Often not aware of alternatives in treatments, trials, and so on • What are the adverse events with this treatment?
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Take a patient profile (which would include both health and demographic information) and compare it to outcomes for other people with similar profiles • Patient access to the lab results or viewer • Compare results and adverse events (AEs) of other people in trial • Allow patients to have access to a calendar of their treatment schedule • Enable patients to enter AEs (for both clinical trials and standard of care) • Allow for at least an aggregate summary of research studies to study participants • Provide consent to have patient contact information made accessible to researchers planning and running trials for which I might be eligible

Patient Advocates

Stakeholder Category	Patient Advocates
-----------------------------	-------------------

Stakeholder Goals	To assist patients in understanding options for treatment, trial participation, potential outcomes and risks
Top Concerns	
Top Needs from caGrid 2.0	<ul style="list-style-type: none"> • Access to Clinical Trials Reporting Program (CTRP), would like to know the stats about trials in their disease • Would like to know how much money is being spent and how successful it's been (provide info on grid) • Consolidate a lot of fragmented information (on trials, treatments, outcomes, AEs) and make it easily available on the grid • Allowing for patient advocate review and comments throughout the protocol development process would be important (including any protocol authoring tools being developed)

4 - Semantic Infrastructure 2.0 Use Cases

4 - Semantic Infrastructure 2.0 Use Cases

February 23, 2011 Working Draft

This section includes the following:

- [4.1 - Translational Medicine, Research and Personalized Medicine](#)
- [4.2 - Life Sciences](#)
- [4.3 - Clinical Trials](#)
- [4.4 - Electronic Health Records](#)
- [4.5 - Terminologies Use Cases](#)
- [4.6 Other Use Cases](#)

4.1 - Translational Medicine, Research and Personalized Medicine

4.1 - Translational Medicine, Research and Personalized Medicine

February 23, 2011 Working Draft

Refer to the Translational Medicine uses cases in section 3 - caGrid 2.0 Use Cases of the current caGrid 2.0 Roadmap (posted with the [caGrid 2.0 Roadmap Documents](#)).

4.2 - Life Sciences

4.2 - Life Sciences

February 23, 2011 Working Draft

Semantic Infrastructure 2.0 needs to address metadata and terminology related requirements from the life sciences domain. This will enable interoperability both between different sub-domains within life sciences, and between life sciences and other domains in caBIG® such as clinical trials and electronic health records.

While life sciences will leverage common semantic functionalities such as Enterprise Conformance and Compliance Framework (ECCF) registry, modeling, forms, behavioral semantics, terminology and value sets, there are aspects specific to the life sciences domain that need to be addressed, including but not limited to:

- High change semantic environment where novel concepts which have not been previously characterized need to be described
- Computational or analytical workflow compositions processing raw data to derive knowledge
- Description of statistical processes related to computational processes to achieve the above
- Working with the platform, enabling semantic description of raw data potentially of large volume (for example next-gen sequencing, imaging data)
- Support of provenance to trace data acquisition and data ownership and also to achieve reproducibility of analytical results
- The caBIO ECCF service specification project on molecular and pathway annotation services from the Integrated Cancer Research (ICR) Workspace

This section highlights some key use cases that depend on data semantics. These use cases provide a representative set to capture the requirements of the life sciences domain. A comprehensive set of all life sciences use-cases can be found at on the [ICRi WG GForge wiki](#). This section includes the following:

- [Discovering a biomarker](#)
- [Finding biomaterial to validate a biomarker](#)
- [Extending the use of a biomarker](#)
- [Exploring predictive power of gene expression in breast cancer metastasis](#)
- [Oncologists in formulating ideas for new clinical studies](#)
- [Multi-Center Ancillary Study in the context of a Consortium Clinical Trial \(extension from Enterprise Use Cases\)](#)
- [Overlay of protein array data on the regulatory pathways with links to patient and cell culture data.](#)
- [Animal model use case](#)
- [An outside researcher requests access to a consortium's Prostate SPORes Federated Biorepositories, eleven instances of caTissue Suite independently maintained and managed](#)
- [High throughput sequencing using DNA sequencing to exhaustively identify tumor associated mutations](#)
 - [Variant: Version A](#)
 - [Variant: Version B](#)
 - [Variant: Version C](#)
 - [Nanoparticle Delivery System](#)
 - [Nanoparticle Delivery System B](#)
 - [Scenario 13](#)
 - [Scenario 13 B](#)
 - [Scenario 14](#)
 - [Scenario 14 B](#)
 - [Scenario 14 C](#)
- [NanoParticle Ontology](#)

**Note**

However, part of the Infrastructure Inception activities include Prototyping [Orchestrations and/or Choreographies](#) (including Life Science workflows) as well as outreach to communities to address other major use cases and requirements. The life sciences communities are engaged with the Roadmap inception efforts now on the following:

- [Lymphoma "workflow" use cases and requirements](#)
- [Dynamic Extensions use cases and requirements](#)
- [Imagining Use cases and requirements](#)

Refer to the pages listed for the use cases and requirements gathering activities. These will be moved to relevant sections in the Roadmaps when mature as use cases, requirements and resulting architecture design.

Discovering a biomarker

A scientist is trying to identify a new genetic biomarker for HER2/neu negative stage I breast cancer patients. Using a caGrid-aware client, the scientist queries for HER2/neu negative tissue specimens of Stage I breast cancer patients at LCCC (University of North Carolina Lineberger Comprehensive Cancer Center-NC Cancer Hospital) that also have corresponding microarray experiments. Analysis of the microarray experiments identify genes that are significantly over-expressed and under-expressed in a number of cases. The scientist decides that these results are significant, and related literature suggest a hypothesis that gene A may serve as a biomarker in HER2/neu negative Stage I breast cancer. To validate this hypothesis in a significant number of cases the scientist needs a larger data set, so the scientist queries for all the HER2/neu negative specimens of Stage I breast cancer patients with corresponding microarray data and also for appropriate control data from other cancer centers. After retrieving the microarray experiments the scientist analyzes the data for over-expression of genes A.

Finding biomaterial to validate a biomarker

In scenario 1, the scientist has validated a biomarker based on available microarray experiments provided by various cancer centers. Now, the scientist would like to request biomaterial in the form of formalin-fixed, paraffin embedded tissue specimens from patients with the appropriate clinical outcomes. The scientist would like to validate the genetic biomarker in a different series of cases, this time using a different technique such as immunohistochemistry. The scientist queries for the presence of appropriate tissue using a caGrid-aware client and for the appropriate contact information of the person(s) responsible for the tissue repository. The scientist contacts the person(s) to begin the protocol for retrieving biomaterials.

Extending the use of a biomarker

The scientist would like to check if genes A could also be used as biomarker for other types of cancer. The flow of events will be similar to Scenario 1 with the exception that the specimen query will not be restricted to Stage I breast cancer patients.

Exploring predictive power of gene expression in breast cancer metastasis

The scientist would like to explore if gene expression patterns can predict how breast cancer will metastasize. The scientist queries all the specimens of breast cancer patients from other cancer centers where their metastasis sites are liver, bone and brain. The scientist then retrieves

the corresponding microarray experiments for these specimens. The scientist analyzes the microarray experiments to explore for a correlation between expression profiles and metastasis sites.

Oncologists in formulating ideas for new clinical studies

The oncologist often wants to first find out the answers to questions such as: How many patients have been seen at our institution with disease x? How does that compare with other institutions? What is the average survival of patients with disease x? How is it different if they are treated with drug x or y? How many patients are there with disease x and TNM stage y at diagnosis? How many patients with disease x relapse after treatment y? This use case is about enabling oncologists to ask these exploratory questions of their clinical databases as well as those at other institutions accessible on caGrid.

Multi-Center Ancillary Study in the context of a Consortium Clinical Trial (extension from Enterprise Use Cases)

The following are the steps in a translational research scenario.

- Within a consortium of cooperating institutions an investigator conducts a search across the clinical data repositories to investigate the feasibility of a potential clinical research idea.
- Within the consortium, the research question is circulated to gauge interest.
- Members of the consortium discuss the research question and approve it as viable.
- The research question is formalized by the coordinating center into a clinical research ancillary protocol for validation of a biomarker as predictive of tumor shrinkage in the context of treatment using an investigational agent and posts to the consortium for consideration (for example, Do patients with a particular marker respond better to treatment with the agent?).
- Consortium member sites choose to join the protocol and agree to accrue patients on to it, collect bio samples from each participant and ship a defined set of bio samples to Central Pathology.
- Participating consortium sites each submit common consent forms, case report forms, and boilerplate Material Transfer Agreements (MTAs) to the appropriate local regulatory offices.
- The protocol meta data, case report forms, standard operating procedures, MTA documents, and other items as needed are finalized and disseminated to each participating site.
- Participants are screened on the basis of eligibility by study coordinator at each site.
- Patients are accrued (by physician or patient self referral) by local staff onto the protocol at each site, and the accrual event is reported to the coordinating center.
- Bio samples are collected and relevant clinical annotations including tumor measurements are collected at the appropriate time points as indicated in the protocol (these are for calculating the primary end point, tumor shrinkage).
- Follow-up appointments are scheduled as specified in the protocol.
- Bio samples are periodically sent to Central Pathology.
- Central Pathology re-labels the samples to hide the source and identity.
- Central Pathology sends out batches of collated bio samples to each of the participating biomarker assay labs.
- A basic scientist at biomarker lab submits the result of biomarker assays.
- Patients are followed for three years from primary treatment date. An annual follow-up visit occurs and a blood sample is taken. Additional clinical annotations are collected.
- The trial closes, and all the data are made accessible to the statisticians.
- A statistician communicates the clinical significance of and evidence for biomarker response prediction.
- A clinical researcher, basic scientist and statistician write a scientific paper reporting the results.
- Data are made available according to funding agencies requirements.

Overlay of protein array data on the regulatory pathways with links to patient and cell culture data.

A clinical research scientist wants to be able to predict the efficacy of tyrosine kinase inhibitors as cancer chemotherapeutic agents. The fact that many oncogenes are tyrosine kinases would predict that such agents should be effective, but several have been synthesized and tested in clinical trials, and the results have been disappointing in the extreme, with more cases of tumor growth stimulation than inhibition. The clinician hypothesizes that these unexpected effects are the result of regulatory feedback loops.

To test this hypothesis, he requires software tools for modeling regulatory pathways. In addition, he needs to determine the state of such pathways in different patients by measuring the state of phosphorylation of the elements (proteins) of these pathways using reverse phase protein arrays. Because the consequences of treating the wrong patient with the wrong agent are so severe, the response of the tumor to the inhibitors

will be tested in vitro, on cell cultures established from tumor biopsies. However, biospecimens and data from those patients who participated in clinical trials of these reagents before their ineffectiveness was appreciated is also available.

Outputs measured on these cultures and biospecimens will include growth rate (determined by flow cytometry or visual counting of cells at different time points, extent of cell death determined similarly, photomicrographs, reports of microscopic observations by trained investigators, rate of DNA synthesis measured by radioisotope or fluorescent labeled precursor uptake and incorporation, and staining with various immune reagents followed by high throughput robotic microscopy and automated image analysis. To develop an understanding that will result in giving the correct drugs to the correct patients, data from the protein arrays will be overlaid on the regulatory pathways and linked to patient and cell culture data.

Animal model use case

The following are the steps describing the scenario.

- A bench scientist chooses candidate glioblastoma genes using human Genome-Wide Association Studies (GWAS), for example, The Cancer Genome Atlas (TCGA).
- The scientist also uses pathway analysis to postulate how multiple "hits" may be involved in tumorigenesis, to direct design of genetically altered mice.
- The scientist uses targeted gene transfer to deliver mutated genes to inbred mice.
- Using inbred mice provides uniform genetic background in which researcher can also investigate mutated candidate genes in conjunction with other gene knockouts.
- The scientist finds mutated gene x expressed in gene y knockout mouse results in glioblastoma development that parallels human pathology.
- The scientist validates that the model reacts in similar way to current therapeutic treatments.
- The scientist uses a mouse model to test new therapeutic treatments, including combinations of drugs chosen to inhibit multiple pathways.
- Clinical scientists use mouse model results to design clinical trials to treat glioblastoma, incorporating genomic information on patients.

An outside researcher requests access to a consortium's Prostate SPORes Federated Biorepositories, eleven instances of caTissue Suite independently maintained and managed

- A Research Fellow at a University has been working at identifying SNPs that might be related to aggressive forms of Prostate Cancer. The Fellow has narrowed the search down to 21 SNPs, and discusses the results with the mentor.
- The mentor just returned from a Bio repository presentation where the mentor learned of the Consortium's federated biobanks built on caTissue, and he mentions that this might be a valuable resource that could aid the research. The mentor suggests that the Fellow contact a colleague at Memorial Sloan Kettering Cancer Center (MSKCC), who is a member of the Consortium.
- The Fellow drafts an email briefly explaining the research and sends it to the MSKCC cancer center member. The Fellow asks about the possibility of searching across the Consortium's federated biobanks for cases that have X and Y and at least 3 years of outcome data. The goal is to collect enough tissue to construct a Tissue Microarray.
- The MSKCC member responds and directs the Fellow to the consortium hub web site where there are details on the policies and procedures for requesting an account to be able to submit a query that would search each of the 11 instances of caTissue Suite. He agrees to be her sponsor.
- The Fellow completes an online form that requests and abstract of the research, the name of the Fellow's institution, the non-profit status of that institution, the name of the sponsoring Consortium member and that person's institution, and a required checkbox indicating that The Fellow has read and agrees to the terms of use.
- The Oversight Committee (OC) of the Consortium Biorepositories has a standing regularly scheduled telephone conference call during which the committee reviews requests to query the federated biorepositories. After each regular call a new set of primary reviewers are elected who will be responsible for thoroughly reading new requests and presenting them to the other members of the OC for a vote.
- The OC has authored a set of appropriate use policy documents against which all requests are measured. The current primary reviewers read the Fellow's application and report to the other members of the OC.
- The OC votes to approve the Fellow's request.
- The Fellow is notified of the OC's decision, and is supplied with an account to the MSKCC instance of caTissue, since this application was sponsored by a member at MSKCC.
- The Fellow logs into caTissue Suite at MSKCC as a researcher, and formulates the parameters of the query. The Fellow submits the query and after a period of time sees a results set that span 8 of the 11 instances of caTissue Suite at the Consortium sites.
- The Fellow uses this information to request tissue from four institutions to build the tissue microarray (TMA).

High throughput sequencing using DNA sequencing to exhaustively identify tumor associated mutations

This is a basic research use case that easily becomes translational when the output of this use case is used, for example, to identify targets for biomarker studies or drug candidates for clinical trials.

Variant: Version A

Version A is "Sequencing of selected genes via Maxim Gilbert Capillary ("First Generation") sequencing." *Nature*. 2008 Sep 4 - *Epub ahead of print* (posted on [GForge for the ICRi workgroup](#)).

1. Develop a list of 2000 to 3000 genes thought to be likely targets for cancer causing mutations.
2. As a preliminary (lower cost) test, pick the most promising 600 genes from this list.
3. Develop a gene model for each of these genes.
4. Hand modify that gene model, for example, to merge small exons into a single amplicon.
5. Design primers for PCR amplification for each of these genes.
6. Order Primers for each exon of each of the genes.
7. Test Primers.
8. In parallel with steps 1-7, identify matched pairs of tumor samples and normal tissue from the same individual for the tumors of interest.
9. Have pathologists confirm that the tumor samples are what they claim to be and that they consist of a high percentage of tumor tissue.
10. Make DNA from the tumor samples, confirming for each tumor that quantity and quality of the DNA are adequate.
11. PCR amplify each of the genes.
12. Sequence each of the exons of each of the genes for each tumor and normal pair of DNA samples.
13. Find all the differences between the tumor sequence and normal sequence.
14. Confirm that these differences are real using custom arrays, the seqenome (Mass Spec) technology and biotage or both. (A biotage is pyrosequencing-based technology directed specifically at looking for SNP-like changes.)
15. Identify changes that are seen at a higher frequency than what would occur by chance.
16. Relate the genes in which these changes are seen to known signaling pathways.

Existing tools for each step are:

1) None; a completely manual process. 2) None; a completely manual process. 3) Data is uploaded from the UCSC Genome Browser to Genboree which has modules for all of the required tasks. 4) Same as 3. 5) Primer3 embedded into a local pipeline developed at the HGSC that keeps primers away from repeats and SNPs. Gaps where this pipeline is unable to create primers are filled in by hand. 6) Manual process. 7) Manual process. 8) It is not known how this was done by the HGSC, but caTissue and similar products can be used here. 9) Manual process. The pathology imaging initiative of Tissue Banks and Pathology Tools (TBPT) might fit in here. 10) Manual process. 11) Manual process. Could a Laboratory Information Management System (LIMS) help here? 12) Software provided as part of the ABI sequencer. 13) Combination of custom, ad-hoc software and manual processes. 14) Manual process. 15) Combination of custom, ad-hoc software and manual processes. 16) Manual process. This *should not be a manual process*, but almost always is, or it is of low quality.)

Variant: Version B

Version B. As above, except globally sequence all genes. *Science* 321: 1807-1812 (2008) (posted on [GForge for the ICRi workgroup](#)). Delete steps 1 and 2 and replace step 3 with: 3) Develop a gene model for each of the genes in the Human genome.

Variant: Version C

Version C. Whole genome sequencing using second generation sequencers. *Hypothetical* (posted on [GForge for the ICRi workgroup](#)).

1. Identify matched pairs of tumor samples and normal tissue from the same individual for the tumors of interest.
2. Have pathologists confirm that the tumor samples are what they claim to be and that they consist of a high percentage of tumor tissue.
3. Make DNA from the tumor samples, confirming for each tumor that the quantity and quality of the DNA are adequate.
4. Sequence each of the sample pairs to the required fold coverage (7.5 to 35-fold, depending on the technology and read length).
5. Map the individual reads to the canonical human genome sequence.
6. Find all the differences between the tumor sequence and normal sequence.
7. Confirm that these differences are real using custom arrays, the seqenome (Mass Spec) technology or biotage or both. (Biotage is a pyrosequencing-based technology directed specifically at looking for SNP-like changes).
8. Identify changes that are seen at a higher frequency than what would occur by chance.
9. Relate the genes in which these changes are seen to known signaling pathways.

Existing tools for each step are:

1) caTissue or similar product. 2) caTissue or similar product pathology imaging tools to be developed by TBPT. 3) caTissue or similar product. 4) Combination of custom, ad-hoc software and manual processes. 5) Proprietary, platform-dependent software, a wide variety of non-caBIG-compatible software packages: Solexa Mapper, Mosaic, 454 Mapper, Velvet Mapper, Solid Mapper (uses a non-standard sequence representation model), Mac. 6) Combination of custom, ad-hoc software and manual processes. 7) Manual process. 8) Combination of custom, ad-hoc software and manual processes. 9) Manual process. This *should not be a manual process*, but almost always is, or it is of low quality.)

Nanoparticle Delivery System

This is a scenario based on finding a nanoparticle delivery system to target a drug which in its free form causes significant side effects. Sorafenib is a Raf kinase inhibitor that disrupts the key Ras/Raf/MEK/ERK cellular pathway that is up-regulated in renal cell carcinoma, glioblastoma multiforme (GBM), and stomach cancer. The drug has significant side effects and a scientist hypothesizes that nanoparticle-assisted targeted delivery of the drug will reduce the required dosing and its side effects.

A scientist interested in targeting this drug to GBM does research on possible nanoparticle-delivery systems that have the following properties:

- Biocompatibility
- Sufficiently long intravascular half-life to allow for repeated passage through and interactions with the activated endothelium
- The ability to have ligands and proteins conjugated on the surface in multivalent configuration to increase the affinity and avidity of interactions with endothelial receptors
- The ability to have functional groups for high-affinity surface metal chelation or radio-labeling for imaging
- The ability to encapsulate drugs
- The capability to have both imaging and therapeutic agents loaded on the same vehicle

Furthermore, the scientist looks for information on nanoparticles that could potentially target the GBM. Integrin-targeted nanoparticles are identified. Synthesis involves ultraviolet (UV) cross-linking of an $\alpha_v\beta_3$ -integrin-targeting ligand attached to diacetylene phospholipids and a cationic lipid. These are sonicated to form polymerized vesicles and the $\alpha_v\beta_3$ -targeted NP can serve as a scaffold for the attachment of therapeutic agents for imaging and therapy.

The physical characteristics have been determined. These include size, zeta potential, and the relevant IC₅₀. In a cell adhesion assay, the 10 of 19 effect of multivalency on IC₅₀ is also measured. Selectivity was also demonstrated in a receptor-binding assay and it is also shown that the $\alpha_v\beta_3$ -targeted NP is not rapidly cleared from the target tissue. Previous studies have shown this particle to be highly stable, to have no measurable toxicity and to specifically target tumor associated vasculature in GBM when conjugated to GFP. Furthermore the particle has been used as an imaging agent when conjugated with Gd³⁺ or Indium²⁺. The $\alpha_v\beta_3$ -targeted NP sorafenib is synthesized. Sorafenib absorption characteristics are available and the concentration of the drug in the system is determined via spectroscopy methods. Other physical properties are characterized.

Nanoparticle Delivery System B

This is the Nanoparticle Delivery System scenario extended. The scientist investigates what data sets are available for in vivo use of the drug. A breast cancer xenograph subcutaneous model is found and cell lines from this system are also available. However, toxicity data for the drug in animal models are not publicly available. The scientist contacts the drug manufacturer and begins in vitro testing. PK/PD in vitro tests, including drug uptake, toxicity and effectiveness, are performed in the model system cell lines, and related and control cell lines by comparing the effects of drug alone, nanoparticle alone, and the combination. Next is in vivo testing with three established animal tumor models. The drug alone, nanoparticle alone, and the combination are administered and tumor size (and other parameters) are monitored. Finally efficacy, dosing, and side effects of the current dosing protocol are compared with targeted nanoparticle delivery of sorafenib.

Scenario 13

This is a scenario based on in vitro profiling of nanomaterial activity. A scientist has created a library of surface-modified nanoparticles with potential as in vivo imaging agents. The scientist would like to use an in vitro approach to gain insight on potential toxicity of these nanoparticles, and exclude those that might be problematic prior to using costly and time-intensive in vivo methods. The mode of administration is considered in selecting a variety of cell types to use in the in vitro assays. Cell cultures are started. Each nanoparticle is added to cultures of each cell type at multiple biologically-relevant concentrations. Multiple cell-based activity assays are used to test each combination of nanoparticle type and cell type, resulting in each nanoparticle being tested in all conditions. Hierarchical clustering algorithms are used to group the nanoparticles based on their activity profiles. Class predictions can be made and verified. Understanding of structure-activity relationships increases, and in vivo correlations among nanoparticles can be tested, and compared with in vitro correlations.

Scenario 13 B

This is Scenario 13 extended. How can an investigator use the dataset described above (and others created in similar ways) to make choices about nanoparticle design to optimize the chance that it would have a favorable in vivo activity? A scientist wants to maximize the circulating half-life of a nanomaterial. One material that has a long half-life is known and the scientist wonders if other nanomaterial compositions have similarly long half-lives.

The scientist would like to look at all available datasets, to see which nanomaterials act similarly to the known agent with a 11 of 19 long half-life. The scientist first queries across cancer center datasets to identify other nanoparticles with the best half-life. Initially, those data sets that use the same experimental protocol and a similar or better half-life are retrieved and compared. Next, the scientist wishes to broaden the search to include data sets that do not explicitly measure half-life, but a common set of cell-based assays. The data sets are normalized and combined. Hierarchical clustering algorithms are used to group the nanoparticles based on their activity profiles across the various cell-based assays. The scientist queries for nanoparticles that cluster closest to the starting nanoparticle with a long half-life, based on their behavior in the cell-based assays. The scientist then tests the hypothesis that the cluster neighbors will also have long half-lives in vivo.

Scenario 14

This is a scenario based on identifying in vivo imaging probes using in vitro cell binding data. The scientist in the previous scenario would like to increase the imaging potential of candidate nanoparticles by modifying them and looking for cell type-specific binding capabilities.

The scientist submits a protocol to the institutional review board (IRB) and begins work upon approval. Libraries of surface-modified nanoparticles with appropriate pharmacokinetic and toxicity profiles are selected and screened for cell binding in vitro using cell cultures of "background" and "target" cell types or classes. The apparent concentration of binding or uptake of each nanoparticle to the different cell classes is measured. Metrics for differential binding to target versus background cells are calculated, and statistical significance is calculated by permutation. (These calculations employ analysis modules available through GenePattern ([posted on GForge for the ICRi workgroup](#))).

To validate the increased specificity for binding target cells, those that provide the best discrimination are further tested ex vivo. Under IRB approval, anatomically intact human tissue specimens containing target and background cells are collected. The tissues are incubated with

nanoparticles and evaluated for nanoparticle localization using microscopy. Further validation is conducted in vivo using an animal model. Animals are injected with the nanoparticle and another tissue specific probe and intravital microscopy is used to determine the extent of co-localization. The scientist contacts the tech transfer office to pursue next steps.

Scenario 14 B

This is Scenario 14 extended, Customizing cell lines to identify nanoparticle probes. Varying the cell lines chosen for the study can help to generate analogous datasets. A scientist wants to find a nanoparticle that targets cancer cells bearing a specific oncogene mutation. Cell assays are performed in multiple cell lines that either do (target) or do not (background) bear this oncogene mutation. The data are analyzed as above to find particles that discriminate between the presence and absence of the mutation. The scientist then tries to validate these probes using independent tumor samples, or in mice genetically engineered to bear tumors that either do or do not express the mutation under study.

Scenario 14 C

This is Scenario 14 extended, Analyzing existing datasets to identify nanoparticle probes. When many nanoparticles have been screened for their uptake in many different cell lines across many cancer centers, a scientist imports all the datasets that involve nanoparticle binding or uptake to cells. The cell lines are reclassified into target or background cells based on a set of criteria (such as tissue type or presence or absence of a oncogene mutation) and an analogous analysis is performed to identify nanoparticles that exhibit differential binding and uptake to different classes of cell lines.

NanoParticle Ontology

This is a scenario based on evaluating and enriching the NanoParticle Ontology (NPO) ([posted on GForge for the ICRi workgroup](#)). The NanoParticle Ontology ([posted on GForge for the ICRi workgroup](#)) is an ontology which is being developed at Washington University in St. Louis to serve as a reference source of controlled vocabularies and terminologies in cancer nanotechnology research. Concepts in the NPO have their instances in the data represented in a database or in literature. In a database, these instances include field names, field entries, or both for the data model. The NPO represents the knowledge supporting unambiguous annotation and semantic interpretation of data in a database or in the literature. To expedite the development of the NPO, object models must be developed to capture the concepts and inter-concept relationships from the literature. Minimum information standards should provide guidelines for developing these object models, so the minimum information is also captured for representation in the NPO.

Nanotechnology is being applied to clinical therapeutics, but this use case could be extended to development of any specialized therapeutics. There are various pre-existing databases holding experimental data that need to be accessible across the entire community to facilitate rational nanomaterial design. Two strategies are being employed. The first is to establish semantic interoperability by finding areas of semantic overlap in the current database models based on controlled vocabularies (NCI Thesaurus, NCI Metathesaurus, Nanoparticle Ontology). The second is to develop a data submission standard based on the extension of standardized models (Biomedical Research Integrated Domain Group (BRIDG), Life Sciences Domain Analysis Model (LS-DAM)) where extensions are supported by controlled vocabularies. New vocabulary is needed to support both of these strategies. New concepts are curated in the controlled vocabularies as appropriate and term definitions are reviewed by the community.

4.3 - Clinical Trials

4.3 - Clinical Trials

February 23, 2011 Working Draft

From the beginning, NCI CBIIT has played an instrumental role in supporting the clinical trial process within NCI, with other agencies and groups in NIH and with other collaborating organizations and companies.

A significant part of this role is providing a foundational basis for these stakeholders to leverage standard terminologies and metadata, and a standards-based framework, in defining semantic entities related to clinical trials such as forms, protocols, and eligibility criteria.

The Semantic Infrastructure 2.0 will expand on these use cases, in order to both facilitate data collection within clinical trials, and enable the information collected to interoperate with other modalities in caBIG® such as the life sciences domain in support of advancing knowledge in disease processes and treatments. NCI CBIIT also plays a leading role in the BRIDG project, which will form a significant basis for the layered approach to metadata and terminology we will adopt in the Semantic Infrastructure 2.0.

This section includes the following:

- [Key Use Cases and Requirements](#)
- [Structured Eligibility Criteria](#)
- [Use Case Decompositions](#)
- [Clinical Decision Support and Clinical Trial Extension](#)
- [Use Case Decompositions](#)

Key Use Cases and Requirements

As a starting point, requirements specific to the clinical trials domain will be collected from the recent Semantic Infrastructure Requirements Elicitation effort, from the NCI CBIIT in-house terminology and metadata curation teams, and from related projects at NCI CBIIT such as the

CTRP, caBIG® Clinical Information Suite, caBIG® Clinical Trial Suite, Janus, and other projects. Requirements are also being collected from external stakeholders including government agencies, standard development organizations (SDOs), organizations and companies such as CDISC (especially the [SHARE project \(cdisc.org\)](#)), and HL7. Community input based on this roadmap document, and the forms and modeling workgroup in the Semantic Infrastructure 2.0 Inception effort, will form another significant source of requirements from the clinical trials domain.

This section highlights some key use cases that depend on data semantics. These use-cases are used as a representative set to capture the requirements of the clinical trials domain. A [comprehensive set of all clinical trials use cases](#) may be found on the CTMS Knowledge Center site.

Structured Eligibility Criteria

When a clinical trial is designed it is done so with a target population in mind. The desired characteristics of the target population are specified during the study design phase and are referred to as the eligibility criteria for the trial. Eligibility criteria for a trial are specified in terms of inclusion and exclusion criteria. If a potential subject meets all of the inclusion criteria and has none of the exclusion criteria then the subject is an eligible candidate for enrollment. Examples of inclusion and exclusion criteria used in eligibility determination follow:

- Hemoglobin ? 8.5 g/dl
- ECOG performance status (PS) 0, 1, or 2

The following is summary of the of the steps:

- PI provides the approved Study Protocol to the Study Registrar.
- Study Registrar selects the study from the list of studies.
- Study Registrar specifies the inclusion criteria as written in the protocol.
- Observations of the subject are made by a healthcare provider, including but not limited to Life expectancy, Karnofsky score, and statements made by the subject in response to questionnaires. These observations are captured in the provider in a clinical form or an EHR system.
- The study registrar leverages the data semantics to identify eligible subjects.
- A PI wants to identify which trials incorporate a test for Alkaline phosphatase.
- The PI wants to correlate this information and derive insights based on the correlation.

Use Case Decompositions

The following is a more refined decomposition of the above use-case:

- Create Eligibility Inclusion and Exclusion Criteria
- Query Eligibility Inclusion and Exclusion Criteria

Clinical Decision Support and Clinical Trial Extension

Clinical trial discovery in the context of the oncology patient undergoing treatment for the primary tumor is time-consuming for the clinician and not often done in the community cancer care realm due to lack of automated access to knowledge of specific ongoing clinical trials. In this scenario, the patient is referred by the primary care physician to an oncologist for an elevated CA 125 tumor marker. The 38-year-old female patient has a past history of breast cancer six years ago with no evidence of recurrence. The patient recently had some abdominal distention which prompted a visit to her primary care physician. An ultrasound had been ordered showing some moderate fluid in the abdomen and pelvis and a solid and cystic mass in the right adnexa measuring 6.8 cm in maximal diameter. Surgical evaluation yields the finding of a stage IIIC epithelial ovarian mucinous cystadenocarcinoma.

The following is a summary of the steps:

- At the time of the referral, the primary care physician used the referral service and attached a CCD summary document of the patient's findings including her imaging study and tumor marker values.
- This was incorporated into the electronic health record along with the assessment of the oncologist and the oncologist's clinical staging.
- The patient's consents were also registered in the electronic health record which included a consent for using the patient's clinical de-identified data to query for a clinical trial match.
- The record is transferred from the EHR to an ECCF knowledge service that evaluates the data against available clinical trials and returns matching trials for the individual patient.

Use Case Decompositions

- An ECCF service continually queries the clinical trials.gov site for up-to-date actively recruiting studies.
- A second ECCF service performs natural language processing and extracts the inclusion and exclusion criteria from the posted active clinical trials
- The caBIG® Clinical Information Suite abstracts a CCD summary that includes diagnoses, age, sex, and types of imaging studies as well as laboratory findings and submits the de-identified CCD to the knowledge management service within the ECCF.
- The knowledge management service develops a profile using the CCD to compare against known active clinical trials and publishes potential trials to the caBIG® Clinical Information Suite.
- The caBIG® Clinical Information Suite links the discovered clinical trials to the patient record based on a unique key and displays this information to the clinician when he accesses that patient's record.

4.4 - Electronic Health Records

4.4 - Electronic Health Records

February 23, 2011 Working Draft

This section includes the following:

- [Introduction to Electronic Health Records Use Case](#)
- [EHR Clinical Forms: Requirements](#)
- [EHR Clinical Forms: Use Case Description](#)
- [EHR: Terminology Requirements](#)
- [EHR: Terminology Use Case](#)
- [EHR: Decision Support Requirements](#)
- [EHR: Decision Support Use Cases](#)
 - [Drug-Drug Interaction Use Case](#)
 - [AJCC Cancer Classification Use Case](#)

Introduction to Electronic Health Records Use Case

The ambulatory oncology electronic health record (EHR) is defined as a set of services and a semantic software platform to compose a "virtual" medical record. This EHR (caBIG® Clinical Information Suite) is being developed by the National Cancer Institute in collaboration with the community cancer centers and will demonstrate the value of computable semantics that enable interoperability and runtime clinical decision support. The caBIG® Clinical Information Suite has several semantic infrastructure requirements and these are highlighted below using a set of high level use cases.

EHR Clinical Forms: Requirements

The [caBIG® Clinical Information Suite](#) is a form driven system for data entry by physicians, nursing staff, and ancillary providers. Changes in the health care landscape across the United States over the last several years has led to new requirements for data structure and semantics. These new requirements have led to the need to deal with "meaningful use" semantics in our form structures. One prominent requirement is the need for any EHR system to use the new semantics in order for providers using these systems to be paid for the care they provide to particular patients. Hence lack of adoption of these data standards could lead to diminished availability for cancer patient care if the physician market shrinks through financial attrition.

Data captured through forms may be re-purposed for business analysis at the institutional level and personal physician level. The data may also be used for adverse event reporting, public health reporting of communicable diseases, and automated reporting to cancer registries. The data may be released for insurance and claims payment and may also be provided to the patient for a longitudinal medical history. Data is exchanged between healthcare providers at the time of referral of the patient for additional care. The caBIG® Clinical Information Suite should also provide data for aggregated and de-identified repositories to support better understanding of healthcare outcomes and best practices.

To serve these many uses, the context and semantics of data entry must be captured and persisted to a backend data store without loss of meaning. To enable this goal of robust data capture, the forms used to capture the data elements must be semantically structured and linked in context using standards-based information models and explicit terminology with traceability through [value set](#) identifiers and coded concept identifiers, allowing aggregation and disambiguation of the captured data.

EHR Clinical Forms: Use Case Description

The EHR Clinical Forms use case has two primary goals. The first goal involves a form designer and construction of a clinical data entry form for use in the graphical user interface of the caBIG® Clinical Information Suite. This form must be semantically consistent and based on HL7 Reference Information Model (RIM) objects, HL7 structural vocabulary, and Office of the National Coordinator (ONC) required code systems for meaningful use. (caBIG® wishes to adhere to regulations and rules from the ONC). The second goal involves the data entry person (a physician, nurse or other health care provider) who defines the value set requirements for the form elements, identifies the rules for skip patterns and form element arrangement, and eventually enters and persists clinical data.

The **form design scenario** aims to enable construction of a form model based on semantic HL7 RIM objects. This allows binding of appropriate code systems and value sets to the form controls of the object, and adding rules for skip patterns and data entry flow. This will allow data capture at the point of care. The forms, their construction objects, and the bound value sets must be persisted for reuse, and the forms must be made available to a user for data entry.

- A Data Entry Form builder gathers requirements for forms from end users who enter data into the system.
- This person builds a form by retrieving form components and value set identifiers from either a repository or the terminology services or from both. The end result of the build process is a semantically aware form and its schema and terminology requirements.
- The form definition is eventually stored on a server. The form metadata is stored in a repository.
- The form definition schema and metamodel are retrieved and a user invokes these to create a data capture form.
- Terminology from the Common Terminology Services (CTS) server is bound at form load time to each control requiring standard terminology.
- The user populates the form with data partially based on retrieved value sets and submits the form and data which is dynamically transformed to a valid Clinical Document Architecture (CDA) document.
- The document is either persisted in a database or exchanged with an external system through a service.

**Note**

A "form control" is any UI object such as a drop down menu, a text field, a submit button. These are all called "form controls".

EHR: Terminology Requirements

The EHR Terminology use case has two primary goals. The first goal involves a data entry form used in the graphical user interface of the caBIG® Clinical Information Suite that requires runtime population of HL7 structural vocabulary, and domain value sets for each coded form control. The second goal involves system requirements for building new value sets when an existing value set is not available for a form control or where large value sets must be subdivided into smaller, nested value sets. Examples of these requirements are the case of localization of geographic value sets or disease specific value sets for new form elements.

EHR: Terminology Use Case

In the graphical user interface **form control terminology binding scenario** the aim is to enable the query of a repository for a form control bound to a form template or HL7 RIM object using a value set identifier, and subsequently to allow the return of an enumerated CTS value set bound as a pick list to the form control. In cases where the form control has not yet been bound to a value set, the system should be able to query by text strings or metadata of the control and return candidate value sets, or when none are returned in this manner, allow direct query of the CTS API for code system concepts to build a new value set.

- When a form is loaded in the caBIG® Clinical Information Suite graphical user interface for the first time, the caBIG® Clinical Information Suite system passes the form identifiers to the repository.
- The repository evaluates the identifiers against known templates or HL7 form objects and passes the identifiers to the CTS service to return all identified enumerated value sets.
- The enumerated value sets are returned and bound to the form controls as pick lists and cached by the caBIG® Clinical Information Suite system for future use.
- Subsequent form loading refreshes the cache by value set versions.

Naïve (initial) form submissions from the Enterprise Conformance and Compliance Framework (ECCF) registry perspective are routed to the form design scenario for value set construction invoking the caBIG® Clinical Information Suite Clinical Form Design use case for control binding to terminology.

EHR: Decision Support Requirements

The caBIG® Clinical Information Suite provides services that cover a wide range of clinical and administrative functionality and that depend on a rapidly changing set of data standards and representations. The system has a set of requirements that include:

- The need to maintain the accuracy and currency of decision support data sources that are dynamic
- The need to provide runtime clinical decision support
- The need to provide just in time software updates that allow interoperability across systems
- The need to interact with systems that can match for clinical trials

Examples of the use cases tied to these requirements are found below.

EHR: Decision Support Use Cases

Drug-Drug Interaction Use Case

A user of the caBIG® Clinical Information Suite starts a patient on a new antiemetic drug. The patient is already on multiple medications, both for treatment of cancer as well as several co-morbid conditions. The user is unfamiliar with a few of the medications the patient is on and the installed EHR has no drug interaction model built in. The user submits the list of medications for the patient to a Semantic Infrastructure decision support service that provides access to runtime drug interaction and contraindication checking.

AJCC Cancer Classification Use Case

A user of the caBIG® Clinical Information Suite uses the 7th Edition of the American Joint Committee on Cancer (AJCC) Cancer Classification system to stage cancer patients. The 8th Edition has now come out and the user would like to upgrade the system to meet the requirements of the state cancer registry. Re-coding the infrastructure for all cancer types in Java is expected to be quite expensive and time consuming. The user queries the ECCF registry for "AJCC Cancer Classification" and finds a plugin reasoner service that implements an OWL version of the AJCC 8th Edition classification system capable of inferring an anatomic stage based on data directly from pathology, imaging, and clinical exam input through a service. The user's system is now able to move with the speed of Cancer Registry requirements rather than the pace of the software vendor.

4.5 - Terminologies Use Cases

4.5 - Terminologies Use Cases

The National Cancer Institute has long been a leader in terminology related services and has provided the cancer community with structured terminology through the NCI Enterprise Vocabulary Services for several years. There is now a much wider range of terminology use cases for the needs from bedside to bench and back and Semantic Infrastructure 2.0 is being designed to address all of these use cases. The use cases require thinking beyond the capabilities of the existing metadata repository as demonstrated in the Cancer Data Standards Registry and Repository (caDSR) and supporting semantic representations of concepts as they are used in clinical information exchange using structured documents. Requirements are also recognized for binding concepts to information models in the abstract as domain values, for use in models such as the BRIDG (Biomedical Research Integrated Domain Group) domain analysis model (DAM) or the life sciences DAM.

As NCI supports these new information models, a fundamental requirement is recognized to continue to support customers who have invested in the common data element capabilities of NCI over the last several years and to provide for customers to continue to use those elements while making the transition to Semantic Infrastructure 2.0. The NCI also needs to continue to support the needs of new customers who are creating information and services that use terminology and structures that exist outside of the BRIDG and LS DAM space. The result will be a more robust representation that captures semantics consistent with usage in clinical care and clinical research, and semantics that will provide more nearly full coverage for the life sciences research community.

Below are several high-level use cases that highlight some of these demanding requirements.

- 1 - Translate local codes to standard code systems
- 2 - Create nested value sets
- 3 - Retrieve semantic code system cross-links
- 4 - Transform CTS 2 value set export to an HL7V3 coded data type
- 5 - Transform an ISO-11179 common data element to an HL7 V3 class object
- 6 - Creation of Value Sets and Value Set Mappings
- 7 - NCI Enterprise Vocabulary Services

1 - Translate local codes to standard code systems

A user has a data base that uses a combination of enumerated values (for example, 0: Male, 1: Female, 9: Unknown) and codes drawn from outside code systems such as Hospital International Classification of Diseases Adapted (HICDA) or Medical Dictionary for Regulatory Activities (MedDRA). The user needs to translate these codes into the codes that are adopted by the Vocabulary and Common Data Elements workspace. The user needs to first determine whether there are existing value sets that already represent the same or a broader conceptual space represented in the user's data base. If one exists, the user then needs to determine whether a mapping exists to the corresponding HICDA or MedDRA codes. If a value set does not already exist, the user needs access to a set of tools that will (a) allow the user to upload the code list along with any corresponding descriptive information, (b) determine which code system(s) are appropriate targets, (c) perform the actual mapping, which is described in a separate use case. Note that the user may need to record local mappings (enumerated values that are of no interest to the larger community) locally while recording mappings between external code systems in a fashion that is accessible by the whole community. Once the mappings are recorded, the user needs access the Semantic Infrastructure 2 framework to locate a service that will allow the user to automate the transformations in the target workflow environment, localizing the service if needed to meet performance or confidentiality requirements and connecting the service to the appropriate mapping(s). The user may also need to embed links to this code in the user's local software.

2 - Create nested value sets

The user is building the specifications for a new application to manage follow-up of chemotherapy patients to track the signs and symptoms associated with a clinical trial of certain chemotherapy agents. The user has been given a set of Common Terminology Criteria for Adverse Events (CTCAE) terms to capture the signs and symptoms but the set of symptoms listed is so large that it does not fit well into a single drop-down menu. The user would like to create subsets of these concepts according related organ systems affected. The user would like to have a unique identifier for each of these sets so they can be reused in future studies. The user has already investigated the available resources and did not find a collection of codes that met these requirements.

3 - Retrieve semantic code system cross-links

A user is constructing a case report form and as part of the form, will have some data entry fields that correspond to laboratory tests that were ordered as part of the study. The user would like to constrain the inputs on the results of these tests to valid possibilities that can be provided via a menu selection. The user would like to submit a list of these laboratory tests to an application that can return an identified value set of possible ranges of answers and associated units of measure for each test. The user realizes that some of the tests will have results that are not available for some reason and would like the appropriate null representations returned as part of the value sets presented by the application.

4 - Transform CTS 2 value set export to an HL7V3 coded data type

A user is building a form that will correspond to a clinical document recording for a pathology report. The user would like the vocabulary drop-down lists to be derived from a CTS 2 value set but would like this value set to be delivered to the document in the syntactical structure required for validation against the CDA schema. In other words, the user would like to submit a list of concepts and have the concepts structured with metadata in a HL7V3 CD data type. The structure that the user anticipates would be delivered as an XML blob with the following structure:

```
<code xsi:type="CD" code="784.0" codeSystem="2.16.840.1.113883.6.42">
```

```
<displayName value="Headache" />
```


</code>

5 - Transform an ISO-11179 common data element to an HL7 V3 class object

A user has a group of caDSR common data elements which have been used in form construction in the past. Now the user would like to move these forms to the HL7 CDA structure. The user finds it possible to represent a portion but not all of the contextual meaning of the data elements in 11179. The user would like to transform the CDEs to HL7 V3 class structures in order to get the full contextual meaning as used in the CDA document. The user would like to submit the list of CDEs to the Enterprise Conformance and Compliance Framework (ECCF) registry transformer and have a set of V3 object classes returned.

6 - Creation of Value Sets and Value Set Mappings

A user has identified one or more sets of permissible values that need to be mapped to standard codes used by the Vocabulary and Common Data Elements workspace. These values need to be assembled in a format such as Excel, XML or simple tab separated values and then be uploaded to a service that will allow them to be mapped to the codes that are endorsed for interchange on the grid. The user must be able to locate existing value sets and to construct new ones where appropriate targets do not exist. The user needs to be able to instruct the service to do automatic first approximation at a mapping, and then needs to be able to search, refine and validate mapping, and to be able to record a "quality" metric that states how closely the mapping actually approximates the intent of the original value. The user may encounter potential problems and omissions during this process and will need to submit suggested changes and enhancements to the appropriate oversight body for potential enhancement. Once the mappings are created, the user needs to be able to download the resulting mappings in an electronic format such as Excel or XML, as well as be able to submit the mappings to a service, either local or centralized, that will represent the mappings via a standardized API.

7 - NCI Enterprise Vocabulary Services

The NCI Enterprise Vocabulary Services (EVS) will provide terminology content and technical support for the Semantic Infrastructure 2.0, and has an extensive user community with use cases and requirements that will help shape development of the new infrastructure. Further details are available on the [EVS Development Path](#) and [EVS - Overview of Use and Collaborations](#) pages.

4.6 Other Use Cases

4.6 Other Use Cases

This section includes the following use cases related to forms:

- [4.6.1 Users create forms to collect data for models and data elements that are already registered in the metadata repository.](#)
- [4.6.2 Create reusable modules](#)
- [4.6.3 Import existing form from caDSR form repository into new form tool](#)
- [4.6.4 Generate fillable form](#)

4.6.1 Users create forms to collect data for models and data elements that are already registered in the metadata repository.

A Data Entry Form builder is notified of a new study or need for a new clinical data capture form by an end user. The Data Manager/Clinical actor begins with the requirements from the end use, usually an existing form and opens a form editor. The actor can create a new Form. The actor can search for an existing model in the metadata registry and drag it into the palette. The pallet self organizes based on the classes and attributes in the model and the existing relations of items in the model. The actor can delete items in the modules, drag attributes into new modules on their form to organize the form according to their anticipated data entry workflow.

If there is not an existing model that the actor can use, the actor browses existing Forms in the local or remote forms registries that could be reused based on search parameters of the form such as the type of study, the type of form (including but not limited to interview, survey, and clinical trial enrollment) or other criteria. If the Data Manager/Clinical Actor is unable to locate an existing form, the actor may search for From Modules in the local or remote Forms registry to construct a new form.

IF the actor finds a form or a module the actor can reuse, the actor drags the existing form or module to the build palette to starts the form project. This is repeated for each section identified by the requirements. When a template is present, the modules self organize according to the template. If a template does not exist, the user connects the modules together with the guidance of the system on the existing classes of the data elements within the modules and relations between them.

If there is not an existing module that the actor can use, the actor can search for possible questions in the metadata repository that meet the questions on their existing form. Search is enhanced by semantic infrastructure to quickly allow users to search for questions by entering text of the question, keywords or values in their value set. The system presents the user with choices the user can put into the forms pallette or temporary area once the user has found all the questions needed to create the form.

The selection of a form, module or question on the palette exposes the attributes of the item in a window where the actor may continue curation.

Selection of the permitted value set is done by matching the semantics of the question in the module to the metadata registry to identify the permitted value set for that attribute. If no value set exists the user is passed to the Common Terminology Services (CTS) service to construct a value set.

If the actor is reusing a class/attribute for which a value set has been specified (in current system at the Data Element Concept (DEC) level), the system shows the actor the value set, existing value domains that have already be paired with the DEC, and the actor can choose to reuse an appropriate value domain, or choose to create a new one by selecting values from the value set.

The actor can optimally choose to search for a new value set by keying in the values from the answer list for the forms question and the system will see if a value set already exists.

The actor adds metadata to the form package and the system checks to see if the questions in the module have been created before with the same value sets, that the system reuses or creates new data element identifiers for each of the form components requiring new identifiers.

The actor can rearrange the permissible values to the preferred order for the form. The actor next defines the rules for skip patterns and form flow logic that is stored with the form. The form is then identified and its associated metadata is passed to the metadata registry. The metadata registry notifies subscribers of the availability of a new form type.

The actor can download an XML description of the form or save the form to a cart for later review or pulling into a data entry system. Software Developer actors can access the form metadata through services to retrieve the form and its associated metadata for implementation in a local system.

4.6.2 Create reusable modules

A Data Entry Form builder is notified of a new study or need for a new clinical data capture form by an end user. The Data Manager/Clinical actor begins with the requirements from the end user, usually an existing form, and opens a form editor. The Data Manager/Clinical actor selects to create a new Module. The actor can search for an existing model, form, module or question in the metadata registry and drag it into the palette.

The actor can chose to have the pallet self organize, or to apply manual organization based on the existing form and workflow. If the actor choes self organize, the questions in the module are organized based on information in the metadata registry related to the questions in the model. If the attribute is found in several models, the actor can chose the model to apply the relations after inspecting the matching models.

The actor can delete items in the modules, and drag attributes into new modules on the form to organize the form according to anticipated data entry workflow.

Selection of the permitted value set is done by matching the semantics of the question in the module to the metadata registry to identify the permitted value set for that attribute. If no value set exists the user is passed to the CTS service to construct a value set.

If the actor is reusing a class/attribute for which a value set has been specified (in current system at the DEC level), the system shows the actor the value set and existing value domains that have already be paired with the DEC. The actor can choose to reuse an appropriate value domain, or choose to create a new one by selecting values from the value set.

The actor can optimally choose to search for a new value set by keying in the values from the answer list for the forms question and the system will see if one already exists.


The actor adds metadata to the form package and the system checks to see if the questions in the module have been created before with the same value sets, that it reuses or creates new data element identifiers for each of the form components requiring new identifiers.

The actor next defines the rules for skip patterns and form flow logic that is stored with the form. The form is then identified and its associated metadata is passed to the metadata registry.

The metadata registry notifies subscribers of the availability of a new form type. The actor can download an XML description of the form or save the form to a cart for later review or pulling into a data entry system.

Software Developer actors can access the form metadata through services to retrieve the form and its associated metadata for implementation in a local system.

4.6.3 Import existing form from caDSR form repository into new form tool

The Data Manager/Clinical Form Builder actor searches the existing repository for a form by name, identifier or keywords. The actor pulls the existing form into the form pallet. The pallet self organizes based on the existing form structure to match to the new form structure (for example, Form to a Form Project, Module to a Common Message Element Type (CMET) ).

The actor can add or delete content from the copied based on need. The actor can reorganize the content based on need.

The actor adds metadata to the form package and the system checks to see if the questions in the module have been created before with the same value sets, that the system reuses or creates new data element identifiers for each of the form components requiring new identifiers.

The actor next defines the rules for skip patterns and form flow logic that is stored with the form. The form is then identified and its associated metadata is passed to the metadata registry.

The metadata registry notifies subscribers of the availability of a new form type. The actor can download an XML description of the form or save the form to a cart for later review or pulling into a data entry system.

Software Developer actors can access the form metadata through services to retrieve the form and its associated metadata for implementation in a local system.

4.6.4 Generate fillable form



Note

This may also be captured elsewhere in the Semantic Infrastructure 2.0 Roadmap.

This use case extends all others. The actor can use the metadata exported from the registry to automatically generate a fillable form that follows all the rules expressed in the form project.

5 - Semantic Infrastructure Functional Requirements

5 - Semantic Infrastructure Functional Requirements

February 23, 2011 Working Draft

The requirements for semantic infrastructure are defined as they relate to the architecture, use cases, and stakeholders. This section presents functional requirements with tracing up to the use cases and down to the service capabilities specified later in this document. This section is not an exhaustive list of requirements and is expected to evolve as additional requirements are analyzed and defined. In addition, Semantic Infrastructure 2.0 will fully support existing caDSR users, including supporting forms created in caDSR.

This section provides a description of the following requirement categories:

- [5.1 Artifact Management](#)
- [5.2 Service Discovery and Governance](#)
- [5.3 Clinical Data Forms Definition and Modeling](#)
- [5.4 Decision Support and Reasoning](#)
- [5.5 Conformance Testing](#)
- [5.6 caGrid 2.0 Platform and Terminology Integration](#)
- [5.7 Other Functional Requirements](#)

The requirements address one or more use cases in each domain, as described in section [4 - Semantic Infrastructure 2.0 Use Cases](#). In addition to the domain-specific use-cases, the requirements also address CBIIT internal development and architecture requirements.

Specifically, CBIIT has standardized on Service-Oriented Architecture (SOA) as the foundational principle for applications architecture and interoperability. CBIIT is currently working to develop a [CBIIT Implementation Guide \(IG\) for the HL7 Service-Aware Interoperability Framework \(SAIF\)](#) which includes the Enterprise Conformance and Compliance Framework (ECCF). This development effort is proceeding in parallel with, and based on, an ongoing dialogue with the caGrid 2.0 and Semantic Infrastructure 2.0 Roadmap projects, to ensure that CBIIT's SAIF IG is consistent with the requirements of the roadmaps. In particular, the CBIIT SAIF IG will contain details on the content, representation, and location within the ECCF Specification Stack for each artifact that will be resident in the Semantic Infrastructure 2.0 ECCF repository and runtime registry. Refer to the [NCI CBIIT SAIF Implementation Guide](#) for more details, as well as a discussion regarding the organizational requirements for supporting computable semantic interoperability and the need to publish formal specifications that can be adopted by external organizations and vendors.

5.1 Artifact Management

5.1 Artifact Management

This section includes the following:

- [5.1.1 Types of Artifacts](#)
 - [Static Models](#)
 - [Behavioral Models](#)
 - [Content](#)
 - [Forms](#)
 - [Specification Content](#)
- [5.1.2 Artifact Management Functions](#)

Artifact management includes support for different formats of models (for example, Unified Modeling Language (UML), Web Ontology Language (OWL), or text), both static and dynamic, as well as the ability to manage content and clinical forms. Artifact management primarily deals with managing artifact lifecycle and authoring of artifact metadata.

A service specification is made up of service metadata, artifacts and the metadata supporting these artifacts. Artifact management enables creating a service specification and helps to accomplish the following:

Improve visibility through publication. When the management service can be integrated into the development, testing and production cycle,

artifacts become available for review and discussion, as well as reference for supporting development. This helps insure proper understanding of applications and services being developed, and provides a standard and controlled method of access.

Annotate artifacts to expand understanding. To further improve the understanding of artifacts, the management service provides the ability to add annotations to both the parts of an artifact (depending on artifact type) and the artifact as a whole. Adding additional semantic definitions to an artifact allows for the searching and location of elements across artifact type, as well as makes clear the intent of a given artifact.

Support governance. When the management service allows for artifact versioning, along with state representation, artifact elements which require governance can be located and interacted with. This functional aspect of the artifact management provides a change history as well as links to external change control systems.

5.1.1 Types of Artifacts

Static Models

Static models include a variety of models with different representations. Static models include but are not limited to:

- Syntactical and semantic models: XML, OWL, RDF representations
- HL7 MIF, UML, 11179 representations
- Meta Models
 - HL7 RIM (Reference Information Model)
 - BRIDG (Biomedical Research Integrated Domain Group)
 - LS-DAM (Life Sciences Domain Access Model)
- Transforms
 - Object Management Group (OMG) Ontology Definition Metamodel Transforms
- Model Constraints
 - Object Constraint Language (OCL), Schematron
- Data Types
 - ISO 21090 and HL7 R2
 - HL7 R1
 - Primitives

Behavioral Models

The behavioral models that will be managed by Semantic Infrastructure 2.0 are identified in the context of the Semantic Infrastructure 2.0 Roadmap, and in compliance with the CBIIT implementation of the Service-Aware Interoperability Framework (SAIF) Behavior Framework (BF).

The SAIF BF defines behavior as "a collection of interactions with a set of constraints on when they can occur in a given Working Interoperability/business process context." The "behavioral models" that will be managed by Semantic Infrastructure 2.0 collectively specify the behavioral semantics of services at the *interface* and *NOT* at the implementation level. Behavior of services provides an unambiguous definition of the service constraints, capabilities, dependencies and interactions. The metadata and grammar required to realize service behavior is called behavioral semantics. Behavioral semantics provide a mechanism for better service discovery and enforcing the constraints at design and runtime.

Dynamic model semantics, as defined in a CBIIT-defined and SAIF-compliant behavioral metamodel, will collectively define the behavioral metadata. A number of technologies still being explored will be used, including UML (Unified Modeling Language) profiles, OWL (Web Ontology Language), Resource Description Framework (RDF), and rules engines (such as Jess), to produce the metadata necessary to support automated (or at least semi-automated) workflow composition. As-yet-unspecified user-friendly tools will be used to provide various contextual inputs for a given workflow including but not limited to known pre- or post-conditions, input data sources, and desired data operations.

Content

Content includes all unstructured text and other forms of content that make up a service specification. Examples include storyboards and scope. Content is an integral part of service specification, and content is leveraged across the enterprise for documentation and communications. Content includes:

- Service specification content, primarily unstructured text
- Images and other representations of static content

Forms

Forms include but are not limited to Clinical Data Interchange Standards Consortium (CDISC), Operational Data Model (ODM), HL7 Clinical Document Architecture (CDA) documents, and HL7 Version 3 RIM-derived forms. This includes all aspects of the document including the style, definitions and semantics. CDISC and NCI CBIIT require a Distributed, **Collaborative** Form Template Development Environment and a Distributed Knowledge Repository to capture and manage its metadata. The following are required:

- Form Templates
- Reusable Form Sections
- Form Definitions

Specification Content

The National Cancer Institute has created many specification documents which include extended datatype flavors for the ISO 21090 datatypes as well as the ECCF specifications for the behavioral framework, information framework, and governance framework. The specifications are an integral part of the semantic infrastructure, allowing the user to fully understand and appropriately apply the many artifacts stored in the ECCF registry.

5.1.2 Artifact Management Functions

Artifact lifecycle management and metadata requirements include the ability to:

- Manage lifecycle, governance and versioning of the models, content and forms
- Establish relationships and dependencies between models, content and forms
- Determine provenance, jurisdiction, authority and intellectual property
- Create representation and views of the information, realized through the appropriate transforms
- Provide access control and other security constraints
- Create annotations for better discovery and searching of artifacts
- Develop usage scenarios and context for the information
- Provide terminology and value set binding
- Provide rules and algorithms for the use of the artifacts in a particular service

The artifacts are bound to the services via the service metadata. The service metadata combined with the artifacts and supporting metadata provide a comprehensive service specification.

The artifact management requirements listed above are derived from the following use cases:

Electronic Health Records: The caBIG® Clinical Information Suite project has adopted ECCF for specifications and Clinical Document Architecture (CDA) documents for interoperability. Project requirements include the need for an infrastructure for managing all the artifacts generated during specification process, including HL7 models and documents. The project also intends to publish these artifacts for the community and vendors. The infrastructure must support better discovery, making all the relevant information available in the right context.

Office of the National Coordinator and other external EHR adopters: ONC has adopted the Continuity of Care Document (CCD) and Continuity of Care Record (CCR) for meaningful use. All national EHR implementations are expected to support forms and the semantics of these forms play a critical role in interoperability. The semantic infrastructure must provide a mechanism to create, store and manage these forms.

Clinical Trials: Clinical trials use forms to capture clinical information, and the semantics captured by these forms are critical for interoperability and reporting. The semantic infrastructure must provide a mechanism to manage the lifecycle of these forms.

5.2 Service Discovery and Governance

5.2 Service Discovery and Governance

This section includes the following:

- [5.2.1 Service Discovery Functions](#)

Service discovery and governance allow service developers to leverage the rich service metadata for better discovery. Service discovery and governance help to accomplish the following:

Promote service reuse: The use of well defined service metadata promotes better discovery and reuse of services during design and run time. Service metadata includes information about service interactions and dependencies. It also includes a classification scheme for organizing services based on business objectives, domain, and usage. It links services to all the supporting artifacts in the specification and provides a placeholder for conformance statements. This enables better reuse across the enterprise and eliminates redundancy.

Establish service policies: Service policies help establish constraints on the service specifications and mandate an approach. Policies can be specified around governance, access control and other design and runtime constraints.

Provide governance: This includes predefined templates, workflows, and governance policies for governing the service lifecycle as well as an approval and review process for service specifications and the ability to promote services through the stages of the service lifecycle.

Enable better discovery: Complex search offers a natural and user-friendly way to find services by progressively refining search results using a variety of criteria including attributes, artifacts, classification, usage scenarios, and dependencies. This includes runtime contract discovery, a powerful query mechanism that allows either the service orchestrator or a program to find the services that best fit the requirements of a given process. This increases both runtime and design time flexibility by enabling selection of services based on computable metadata.

5.2.1 Service Discovery Functions

Service discovery functions include the ability to:

- Identify the service endpoint for analysis
- Identify the service directory endpoint for analysis
- Extract the service interface
- Annotate the service interface providing undiscovered features or behaviors
- Manage lifecycle, governance and versioning of the service interfaces

The requirements listed above are derived from the following use cases:

Electronic Health Records: The caBIG® Clinical Information Suite project is developing service specifications and lacks the infrastructure to govern these services. Vendors and external implementations are expected to leverage the caBIG® Clinical Information Suite service specifications and there is currently no infrastructure that allows easy discovery and consumption of this information.

CBIIT Projects: CBIIT has adopted SOA. Service lifecycle management and governance are industry best practices for all organizations adopting SOA. Better service discovery and reuse improves productivity, avoids redundancy and makes it easier for the CBIIT enterprise architecture governance team to manage NCI's enterprise services portfolio.

Life Sciences: Service discovery based on a rich metadata and semantics of the underlying data play a critical role in developing research pipelines. Research pipelines are developed by connecting data and analytical services together to achieve a research objective.

Other national initiatives: All EHR vendors and national initiatives rely on a services paradigm for integration and interoperability. A standardized services metamodel makes it easier for participating organizations to discover and reuse services.

caGRID 2.0 Platform: The caGRID 2.0 Platform provides a runtime registry for service discovery. This service registry relies on a small subset of information for discovery. The semantic infrastructure provides a mechanism to leverage rich service and artifact metadata to extend this capability.

5.3 Clinical Data Forms Definition and Modeling

5.3 Clinical Data Forms Definition and Modeling

This section includes the following:

- [5.3.1 Clinical Data Form Functions](#)

Clinical Data Forms Definition and Modeling

Clinical Data Forms are the primary channel for capturing information in the healthcare and clinical domain. Forms also play a key role in information exchange and are critical to supporting interoperability in healthcare.

A form differs from a document, in that a document is used to capture information, while a form defines skip patterns, validation rules, and other aspects required to capture or render information for a document.

A document in this context is specifically a clinical document which represents information about a clinical activity. The document contains the specific information gained during that clinical activity and supports the broader definitions of a document. Documents can be transformed into human readable forms, and be transferred or transmitted electronically for use across different systems.

Clinical data forms definition and modeling help to accomplish the following:

Define data entry forms using robust data representation. Ultimately the data that is captured on a form is used in many ways, but that data must provide a high level of meaningful use to insure the consumer knows how the data was captured and what context it represents. In this way even a simple question on a form may result in a much more complex representation in the data. As an example, a Yes or No question on a form may result in a codified representation of an observation.

Reuse contextual representation. Since a given form may collect data for a context that might be common to many forms, being able to reuse these elements in a way that insures contextual consistency is a must. Forms created with the form definition tool must retrieve from well defined metadata sources that provide common contexts, default values, and coded representations including value set binding.

Reuse form elements. When defining a form element which is bound to a specific contextual representation, it should be easy to reuse that element with minimal reconfiguration.

Provide governance support. Forms and the supporting schemas need to be versioned as well as support the governance workflows. This insures that documentation follows a consistent and planned use.

5.3.1 Clinical Data Form Functions

The functions of clinical data forms include the ability to:

- Define model objects for reuse
- Define form templates
- Bind value set to data element
- Provide default form delivery
- Provide form data transformation

Based on the use cases the key forms requirements include:

- Tools and services for defining form templates
- Ability to leverage models and reusable segments for defining these forms

- Ability to bind terminology in the form of value sets to form controls
- User friendly tools that hide the complexity of the underlying semantics

The requirements listed above are derived from the following use cases:

- *Electronic Health Records*
- *ONC and Other external EHR adopters*
- *Clinical Trials*

5.4 Decision Support and Reasoning

5.4 Decision Support and Reasoning

This section includes the following:

- [5.4.1 Decision Support Functions](#)

Decision Support and Reasoning

One of the primary reasons for having structured data is to provide the ability to automate decision support and reasoning across information models, data types, and the terminology associated with the attributes of each data type. For the ECCF registry to provide maximal value to end users, it is necessary to support common decision support functions across the enterprise and to extend that through services to the end users. In effect the semantic infrastructure must provide the tools to support Decision Support solutions:

Identify sources of valued information. Using the semantic metadata as a source, reasoning systems need to be able to identify the sources of information which are key to a given decision support solution. The services, models, and annotations provide definitions which can identify candidate sources for integration.

Common representations and transformations. To make decision support services viable, it is necessary that information be consistent and provide the ability to transform data for use in various tools and reasoning solutions.

Support for classification. The system provides for data classification, discovering new knowledge about key elements. This classification process is based on description logic and business rules which process the semantic structures of artifacts. Classification information should be added to the pool of knowledge about given structures and related information

Support for expert system rule processing and choreography. Using systems such as the OWL classifiers (Pellet, Fact++, Hermit), rule based expert systems (Jess, Drools), and work with RDF (Resource Description Framework) choreography languages (SPIN), the decision support system should be able to be applied in a choreographed layered fashion. Key to this process is a choreography engine which matches data with rules and a reasoning environment. Because of the complexity of the reasoning requirements, the OWL 2 specification is required in order to support the Semantic Infrastructure 2.0 requirements.

Integration with service registries. Since the artifact metadata provides definitions of data, the service registry provides the data access needed to process information. If a given artifact is a service, the decision support system determines the necessary definitions to integrate a service into decision support for the gathering of data.

The requirements listed above are derived from the following use cases:

- *Translational Medicine*
- *Research and Personalized Medicine*
- *Life Sciences*
- *Terminologies Use Cases for Clinical Trials*
- *Electronic Health Records*

5.4.1 Decision Support Functions

Decision support functions include the ability to:

- Query artifact metadata to locate useful artifacts for decision support
- Query service metadata to locate services matching artifacts and metadata definitions
- Create a decision support definition
- Create a decision support session
- Provide scheduling and access information to choreographer
- Select rules and rule system environment
- Execute reasoning systems against gathered data providing additional insight.

The requirements listed above are derived from the following use cases:

- *Electronic Health Records*
- *Clinical Trials*

5.5 Conformance Testing

5.5 Conformance Testing

Services specifications developed by NCI and the community have to be testable to ensure that the implementation conforms to the specification. Conformance testing leverages the artifact and service registries along with predefined reasoning systems to validate that an implementation adequately addresses the requirements stated in the service specification. An example of service requirement is the ability to specify a response time in the specification (design time) and validate that this response time is adequate for an implementation of the service. Additional test points include but are not limited to binding to specific terminologies and domain models.

Conformance testing allows both CBIIT and other HL7 SAIF adopters to validate specifications as follows:

Analyze a given artifact for its stated ECCF purpose. Determine if a given artifact satisfies the requirements of the ECCF artifact that it declares itself to be. This analysis should look at such things as datatypes matching the appropriate level (abstract data types in a Platform Specific Model (PSM)).

Analyze a given artifact to verify traceability. Determine if a given artifact provides correct traceability from level to level. The analysis should look at naming conventions and stereotypes to determine correctness along with promotion of data types from different levels of abstraction.

Analysis of accessibility and interoperability. Used to determine if a given service matches its proposed service specification. Also determine if an artifact or specification is complete as it relates to data binding and value set binding.

Conformance Testing Functions include the ability to:

- Analyze an artifact for ECCF Conformance and traceability
- Produce a non-conformance statement
- Interact with governance systems

The requirements listed above are derived from the following use cases:

CBIIT's adoption of ECCF: ECCF requires all specification developers to make conformance statements; the conformance testing framework leverages these conformance statements to generate validation tests.

Other national initiatives: Other national organizations like NIST are adopting a similar approach to conformance testing.

5.6 caGrid 2.0 Platform and Terminology Integration

5.6 caGrid 2.0 Platform and Terminology Integration

This section includes the following:

- [5.6.1 Service Generation](#)
- [5.6.2 Service Discovery and Utilization](#)
- [5.6.3 Service Orchestration and Choreography](#)
- [5.6.4 Policy and Rules Management](#)
- [5.6.5 Event Processing and Notifications](#)
- [5.6.6 Data Representation and Information Models](#)
- [5.6.7 Data Management](#)
- [5.6.8 Data Exploration and Query](#)
- [5.6.9 Provenance](#)
- [5.6.10 Data Semantics](#)
- [5.6.11 External Data Repositories](#)

caGrid 2.0 Platform and Terminology Integration

The semantic infrastructure has to support seamless integration with the caGrid 2.0 platform. The following are some high-level platform and terminology requirements that are either supported or addressed by the semantic infrastructure.

5.6.1 Service Generation

Service generation is the ability to generate services from user defined service metadata. The semantic infrastructure provides this metadata and the platform leverages this metadata for service generation. The constraints and policies specified in the semantic infrastructure are inherited by the platform and are enforced as runtime policies.

Additional platform specific and runtime information is provided by the developer at the time of service generation.

5.6.2 Service Discovery and Utilization

This group of requirements focuses on enabling developers of composite services and applications to discover, compose, and invoke services.

This includes the discovery of published services based on service metadata and the generation of client APIs in multiple languages to provide cross-platform access to existing services.

Discovery includes service discovery, data discovery, and policy discovery. Service discovery allows primary users as well as secondary users to locate a service specification and instances based on attributes in the service metadata (for example, via a search for specific microarray analysis services). Data discovery enables secondary users to find the types of data available in the ecosystem as well as summary-level information about available data sets. Policy discovery allows application developers to find and retrieve policies on services.

The platform will use the semantic infrastructure service metadata to address all the service discovery requirements. The semantic infrastructure relies on metadata about services and artifacts.

Link to use case satisfied from caGrid 2.0 Roadmap: As institutions share de-identified glioblastoma data sets, they are available to others via data discovery. The treatment recommendation service used by the oncologist is able to discover these new data sets and their corresponding information models, and include that data for subsequent use in recommendation of treatment.

Link to use case satisfied from caGrid 2.0 Roadmap: all of the data management and access services in the use case are utilized by application developers to build the user interfaces that the clinicians use during the course of patient care.

5.6.3 Service Orchestration and Choreography

Service orchestration and choreography allows both application developers and non-developers to discover service "building blocks" that can be composed dynamically to provide business capabilities. Special cases include the orchestration of multiple services for a distributed query, or for a transactional workflow. Service orchestration and choreography will leverage static and behavioral semantics from the Semantic Infrastructure 2.0.

The semantic infrastructure provides the behavioral semantics required for dynamic composability of services or generation of distributed queries. This includes runtime contract discovery and negotiation to determine composability of services based on service capabilities and constraints.

Another use case is dynamic retrieval and enforcement of the policies that are in effect for a service interaction in the areas of logging, validations, data transformation, or routing. This information can be used either during the design of the orchestration or during the execution of the defined flow.

Link to use case satisfied from caGrid 2.0 Roadmap: Federated query over The Cancer Genome Atlas (TCGA) data and other data sets is performed using a service orchestration.

5.6.4 Policy and Rules Management

Policy and Rules Management allow non-developer secondary users to create policies and rules and apply them to services. The scope of policies includes, but is not limited to, definition and configuration of business processing policy and related rules, compliance policies, quality of service policies, and security policies. Some key functional requirements for managing policies include capabilities to author policies and store policies, and to approve and validate policies and execute policies at runtime.

The semantic infrastructure will provide a mechanism to specify policies, including business processing policies and related rules, compliance policies, and quality of service policies. Tools and services for creating security specific policies will be provided by the caGrid 2.0 platform and will be used by the semantic infrastructure. All other policies specified in the semantic infrastructure will be enforced by the platform at runtime.

Link to use case satisfied from caGrid 2.0 Roadmap: Each institution has different data sharing needs, access control needs, and business rules for processing that are defined and customized. For example, policy at the pathologist's institution may state that the patient is scheduled for a visit when the review is complete.

5.6.5 Event Processing and Notifications

Event Processing and Notifications enables monitoring of services in the ecosystem and provides for asynchronous updates by services, effectively allowing a loose coordination of services that both provide and respond to conditions (possibly defined in business rules).

The semantic infrastructure will provide a placeholder to specify events and triggering conditions for data and services. The platform monitors these events at runtime and acts on these events.

Link to use case satisfied from caGrid 2.0 Roadmap: As patient care proceeds, the system notifies the designated clinicians that data (for example, images) are ready for review. Similarly, when notifications are received, event processing logic allows the appropriate parties to assign clinicians for care. In order to facilitate better treatment (a learning healthcare system), as new de-identified glioblastoma data is made available, notifications are sent that could indicate a recommended change in the treatment plan.

5.6.6 Data Representation and Information Models

This set of requirements includes providing an application developer with the ability to define application-specific attributes (for example, defined using ISO 21090 healthcare datatypes) and an information model that defines the relationships between these attributes and other attributes in the broader ecosystem. In particular, the last requirement suggests linked datasets, where application developers can connect data in disparate repositories as if the repositories are part of a larger federated data ecosystem. Additional requirements include the ability to publish and discover information models. Support is needed for forms data and common clinical document standards, such as HL7 CDA. To support the use of binary data throughout the system, the binary data must be typed and semantically annotated.

All information models, their representation and binding to datatypes and terminologies will be managed by the semantic infrastructure. The ability

to publish and discover information models will be supported by the semantic infrastructure, and the platform will leverage these capabilities.

Link to use case satisfied from caGrid 2.0 Roadmap: The pathology, radiology and other data have various data formats which must be described, and the information model for the patient record must link between these various datatypes. The complete information model includes semantic links between datasets to build a comprehensive electronic medical record. Annotations on data are defined and included in the information model.

5.6.7 Data Management

Data management includes linking of disparate data sets and updates of data across the ecosystem. Data updates may include updates to multiple data sources, necessitating the need for transactions.

Linkages between the different disparate data sets will be managed by the semantic infrastructure. Data updates that trigger transactions are captured by the platform and are propagated upstream to the semantic infrastructure. An example would be the platform monitoring events to identify changes to data.

Link to use case satisfied from caGrid 2.0 Roadmap: the patient has an electronic medical record that spans multiple institutions. The clinical workup data (for example, genomics and proteomics data) is linked to the clinical care record; similarly pathology and radiology findings must be attached to the patient's electronic medical record.

5.6.8 Data Exploration and Query

The wealth of data must be accessible, resulting in the need for exploration of available datasets. This includes the ability to view seamlessly across independent data sets, allowing a secondary user to integrate data from multiple sources. In addition, the query capability must support sophisticated queries such as temporal queries and spatial queries.

The semantic infrastructure will provide metadata for discovery of these datasets. Complex temporal and spatial queries will be informed by the metadata but will be formulated and executed by the platform.

In order to also discover dataset contents exposed on the grid, the ECCF registry must have linkages from dataset metadata to the metadata about the data they contain. This is distinct from the metadata about the dataset (the owner, creation time, table structure of fields and attributes) and instead describes the type of data contents of the dataset so that a user can retrieve portions of a dataset of some type.

Link to use case satisfied from caGrid 2.0 Roadmap: The oncologist must be able to quickly find glioblastoma data sets, indicating the fields that he is interested in comparing from his clinical data in order to find similar disease conditions and associated treatment plans. Temporal queries allow clinicians to identify changes in patient condition and treatment over time.

5.6.9 Provenance

Provenance encompasses the origin and traceability of data throughout an ecosystem. This is a clear requirement directly from the use case in order to ensure that all steps of patient care and research are clearly linked via the patient record.

The semantic infrastructure will provide data provenance support.

Link to use case satisfied from caGrid 2.0 Roadmap: The origin of data is tied to the data creator, allowing the oncologist performing the match against TCGA data and other datasets to include and exclude data sets based on their origin.

5.6.10 Data Semantics

In a diverse information environment, semantics must be used to clearly indicate the meaning of data. This requirement is expected to be addressed by the semantic infrastructure, although there will be a touchpoint between caGrid 2.0 and Semantic Infrastructure 2.0 to annotate data with semantics. Integration with the semantic Infrastructure will enable reasoning, semantic query, data mediation (for example, ad hoc data transformation) and other powerful capabilities.

Data semantics are captured in the semantic infrastructure and the platform will leverage the semantic infrastructure interfaces for reasoning and analysis.

Link to use case satisfied from caGrid 2.0 Roadmap: The oncologist accesses the TCGA database to search for de-identified glioblastoma tumor data that is similar to the patient data exported from the hospital medical record. During this search, the semantics of the data fields are leveraged to indicate matches between TCGA data fields and the hospital medical record data fields.

5.6.11 External Data Repositories

There are numerous data repositories on the web today. These data repositories contain essential information that must be accessible to services in the ecosystem. As a result, caGrid 2.0 must provide capabilities to integrate these external repositories into the grid with the assumption that the remote service cannot be changed.

The semantic infrastructure will support integration with other metadata repositories, allowing the platform to leverage the semantic infrastructure for federated metadata discovery and analysis. The federated data query capabilities will be implemented by the platform.

Link to use case satisfied from caGrid 2.0 Roadmap: The oncologist searches both TCGA glioblastoma data as well as de-identified data that has been added by care providers around the country. The additional data sets are external data repositories.

5.7 Other Functional Requirements

5.7 Other Functional Requirements

This section lists other functional requirements from the community that have not been directly traced up to a use case. Service capabilities have not been derived from these requirements at this time. These requirements are part of the Semantic Infrastructure 2.0 Roadmap and will be prioritized according to strategic objectives of CBIIT and resource availability. The following areas are included:

- [Tools Artifact Search and Access](#)
- [Tools Artifact Authoring](#)
- [Artifact Governance and Lifecycle Management](#)
- [Artifact Analysis](#)
- [Search and Access Services](#)
- [Administer Services and Specifications](#)
- [Analyze Services](#)
- [Plug-ins, Loaders and Miscellaneous Development Tools](#)
- [Forms Editor](#)
- [Knowledge Manager](#)

Tools Artifact Search and Access

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Search for content to include in a model or form by leveraging the existing semantic structure of the ISO 11179 data elements to find related or similar items.

Download data element and form artifacts in various formats: XML, Excel, Resource Description Framework (RDF).

Retrieve a model in UML, XML/XMI, or Web Ontology Language (OWL) format.

Tools Artifact Authoring

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Create ISO 11179 data element from concepts and Value sets in Reference Information Model (RIM) or other terminology.

Add metadata to existing item to reflect context of use, such as alternate names or definitions, reference documents.

Ability to add or modify assertions of equivalence between items such as asserting that an existing common data element (CDE) is equivalent to a RIM-based data element.

Extend existing models.

Add semantic annotations to services.

Constraining and extending models will be in accordance with user defined business rules.

Artifact Governance and Lifecycle Management

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Allow curators to set the status of forms according to user defined business rules.

Allow curators to set the status of data elements according to user defined business rules.

Perform impact analysis when changing a data element, value set or model to see where it is used and determine if other changes to other artifacts are needed.

Artifact Analysis

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Support comparison of models in easy to use user interfaces.

Support comparison of value sets and alignment of like items.

Support comparison of forms so users can compare items side-by-side (Question by question).

Support comparison of ISO 11179 data elements.

Search and Access Services

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Search for a service by finding those that contain particular models or parts of models (classes, associations, data elements, value sets or concepts) by name or identifier.

Administer Services and Specifications

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Annotate service descriptions with data and service semantics to support search and discovery across services, such as finding a service that supports registering a patient on a clinical trial, or creating a new protocol.

Validate conformance to a particular model including analysis of whether class definitions, associations and concept annotations are consistent with the base model.

Extend existing services to include newly created model, forms or data elements.

Analyze Services

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Execute service queries to find services that can be combined, where the output of one service is an appropriate input to another.

Execute service queries to find services that transform data from one format into another input format matching that required for use with another service.

Plug-ins, Loaders and Miscellaneous Development Tools

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Support loading of data elements and value sets from Excel spreadsheets and ensure there is no duplication with existing content.

Support deriving data elements and value sets from Excel spreadsheet data.

Forms Editor

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Allow curators to create and manage collections of downloadable forms.

Export forms in Clinical Document Architecture (CDA) format.

Export forms in Operational Data Model (ODM) format.

Export forms in caDSR-compatible format so that vendors who have made their product compatible with caDSR forms can still use caBIG® forms in their systems.

Import forms from caDSR format so users can add RIM-derived data elements to existing forms.

Allow curators to organize questions on forms and value sets according to the workflow defined by an existing institutional form.

Allow curators to copy forms to create new form content.

Allow curators to reuse Question/Answer sets from form to form.

Allow curators to reuse and manage modules independently of forms.

Allow curators to download forms in XML or Excel format.

Allow curators to specify default values for questions.

Allow curators to specify repeating groups of questions.

Provide a platform-independent export of the form so that it can be consumed by other forms software.

Include the meaning of each of the values in the value set on the form.

Allow curators to record what token is being persisted in the data for a particular value set which could be different from the values in the value set, for example record a Medical Dictionary for Regulatory Activities (MedDRA) Code associated with a particular Common Terminology Criteria for Adverse Events (CTCAE) term.

Provide notification when viewing a form if the data element or value set has changed since the form was created.

Allow curators to monitor changes to forms.

Provide search for forms that use a particular model.

Provide search for forms that use a particular value set.

Provide search for forms that use a particular value or set of values.

Provide search for forms that use a particular data element.

Provide search for forms that use a particular set of class associations.

Provide search for forms that use a particular module.

Provide search for forms that collect data dealing with a particular RIM class or terminology concept.

Provide the ability to support complex ISO 21090 datatypes on forms so that the values needed to populate the attributes of the datatypes can be captured.

Provide support for questions on forms that are based on a calculation or derivation from other questions on the form.

Provide information and ability to search Forms based on the form owner, dates created or modified and by whom modified.

Provide search for forms using a particular class or set of classes.

Provide the ability for curators to search for forms in all workflow statuses, but hide draft content from the casual user (for example, ability to "Publish" or "Not Publish" forms).

Provide the ability to search for forms that are part of a particular protocol.

Provide the ability to search for forms that pertain to a particular trial type.

Provide the ability to print the form.

Provide the ability to view the form as it will look when rendered.

Provide the ability to attach reference document to the form.

Provide the ability to include form, module and question instructions on the form.

Provide the ability to specify whether form contents are mandatory, optional or conditional.

Provide the ability to author new question text or use standard question text.

Provide the ability to search for forms that pertain to a disease.

Provide the ability to transform spreadsheet data into RDF stores.

Knowledge Manager

[Link to related Semantic Infrastructure 2.0 Roadmap page](#)

Provide the ability to transform models represented in ISO 11179 into RDF stores.

Provide the ability to transform relational databases into RDF stores.

6 - Semantic Infrastructure 2.0 Architecture

6 - Semantic Infrastructure 2.0 Architecture

February 23, 2011 Working Draft

This section includes the following:

- [6.1 - Overview of Semantic Infrastructure 2.0 Architecture](#)
- [6.2 - Overview of Semantic Infrastructure 2.0 Capabilities and Services](#)
- [6.3 - Tools for Semantic Infrastructure 2.0](#)

- 6.4 - Tie-in with Terminology and Platform

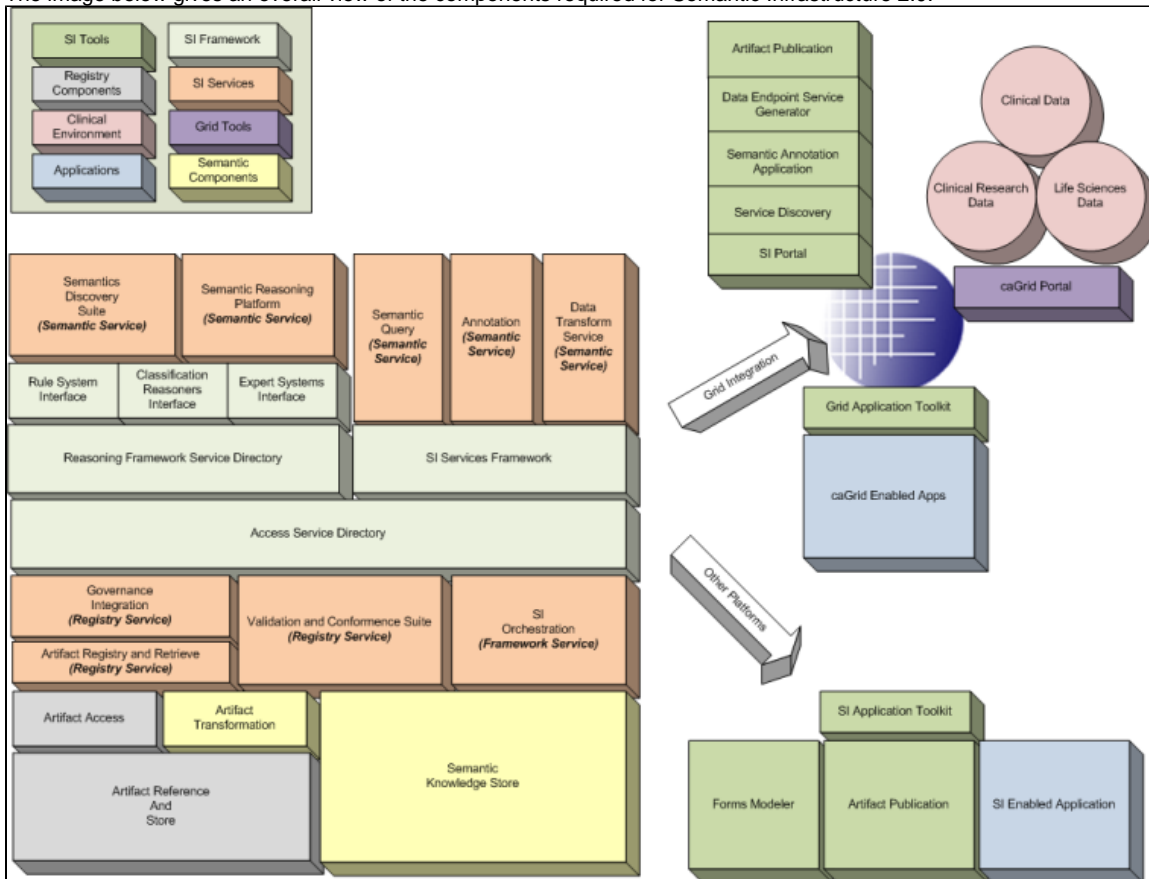
6.1 - Overview of Semantic Infrastructure 2.0 Architecture

6.1 - Overview of Semantic Infrastructure 2.0 Architecture

February 23, 2011 Working Draft

This section provides an overview of the proposed architecture, which includes a set of core services and tools. Section [6.2 - Overview of Semantic Infrastructure 2.0 Capabilities and Services](#) summarizes the profile of the solution with mapping to appropriate requirements and use cases. Section [6.3 - Tools for Semantic Infrastructure 2.0](#) provides an end user's view of the tools. Section [6.4 - Tie-in with Terminology and Platform](#) describes integration with the platform and terminology.

The image below gives an overall view of the components required for Semantic Infrastructure 2.0.



Component Name	Description
Registry Components	Refer to the descriptions for each component in the following rows.
Box Name: Artifact Reference and Store	This registry component is a store or registry that contains references to the various artifacts. Each artifact should have a URL that can be used to physically access the file. Each artifact reference is accompanied by a checksum or some other method to be able to verify the accessed object.
Box Name: Artifact Access	This registry component provides programmatic access to artifacts in the Artifact Reference and Store.
Box Name: Artifact Registry and Retrieve	This registry service provides a programmatic interface for interacting with the artifact reference registry.

Box Name: Governance Integration	This registry service provides state mechanisms about known artifacts that can be accessed and reviewed through governance activities.
Box Name: Validation and Compliance Suite	This registry service integrates with the reasoning system to validate the compliance of specific artifacts (ECCF models).
Semantic Components	Refer to the descriptions for each component in the following rows.
Box Name: Semantic Knowledge Store	This semantic component provides a physical representation of semantics that have either been derived through artifact analysis, or through manual annotation. This store could be represented by an RDF (Resource Description Framework) triple store.
Box Name: Artifact Transformation	This semantic component provides a function that takes as input some artifact and provides output in alternative representations. This might include a class model in UML being transformed to an OWL ontology.
Box Name: Semantics Discovery Suite	This semantic service takes as input artifacts or artifact transformations and extracts as many semantic representations as possible. The details of the semantics will depend on artifact type, representation, and completeness. The results are then stored in the Semantic Knowledge Store.
Box Name: Annotation	This semantic service provides functionality that allows additional semantics to be added about an artifact reference in the Semantic Knowledge store, and is used to augment the semantic representations which were automatically discovered.
Box Name: Data Transformation Service	This semantic service provides a set of transformation functions which are designed to transform data; this may include transforming data graphs into CSV, result sets into XML, or other reasonable transformations. This function may use semantics stored about artifacts to aid in the transformation function.
SI Framework Components	Refer to the descriptions for each component in the following rows.
Box Name: Access Service Directory	This framework component provides the set of services that are available within a Semantic Infrastructure implementation which are designed to manage artifacts. This will allow for the coordination of stores and services across the grid.
Box Name: Reasoning Framework Service Directory	This framework component provides the set of services that are available within a Semantic Infrastructure implementation that provide reasoning functionality to analyze artifacts and instance representations of associated data.
Box Name: Rule System Interface	This framework component provides integrations of one or more rule systems to support to the Semantic Infrastructure in expressing business rules and behaviors.
Box Name: Classification System Interface	This framework component provides integrations for one or more classification tools. These tools are systems that process semantic and dependent information to determine relationships and associations of classes and individuals which may be expressed in an artifact, its annotated information, or instance representations of associated data.
Box Name: Expert System Interface	This framework component provides integration to one or more expert systems. These systems utilize a set of known facts and domain expert definitions to determine additional semantics and functional definitions within the artifact semantic information and instance representations of associated data.
Box Name: SI Services Framework	This framework component provides interface support to semantic and reasoning services.
Box Name: Orchestration	This framework service manages the internal flow of operations that can be performed. This includes automating the transformation and semantic discovery and the utilization of various rule systems or classification systems.
Integrations and applications	Refer to the descriptions for each component in the following rows.
Arrow: Grid Integration	The grid integration represents the interaction of Semantic Infrastructure services with the caGrid
Box Name: Grid Application Toolkit	This Semantic Infrastructure tool provides libraries and functions that ease the creation of new caGrid enabled applications. This tool kit will provide a method to integrate caGrid 1.0 applications to ease applications into the caGrid 2.0 environment.
Box Name: caGrid Enabled Applications	caGrid enabled applications include any application written to the caGrid specification.

Box Name: caGrid Portal	This caGrid application is a tool for accessing aspects of of the caGrid in a partner site.
Box Name: Clinical Data	This represents clinical information that may be exposed to the grid. Using the portal, an authorized user may expose data or services onto the grid; this might include outcome markers, treatment plans or other relevant information
Box Name: Clinical Research Data	This represents clinical research data that might be exposed to the grid. Using the portal, an authorized user may expose data or services onto the grid; this might include trial cohort qualifications, raw data, or publishable results.
Box Name: Life Sciences Data	This represents life sciences data that might be exposed to the grid. Using the portal, an authorized user may expose data or services on the grid; this might included gene array studies, algorithms, methodologies and data sets.
Box Name: SI Portal	This application provides a user interface for implementations of the Semantic Infrastructure framework components. The user would use this tool to access the functionality of the Semantic Infrastructure components exposed on the grid. Probably a part of the caGrid Portal
Box Name: Service Discovery	This tool and portal component provide a user with the ability to enter key words and tags or semantic queries to help determine the locations of artifacts and communication endpoints.
Box Name: Semantic Annotation Application	This tool and portal component provides a user with the ability to annotate artifacts and communication endpoints to help the user perform queries.
Box Name: Data Endpoint Service Generator	This tool allows a user to quickly create a data endpoint and make it available on the caGrid, merging the data source with a SPARQL Endpoint and structuring for access.
Box Name: Artifact Publication	This tool allows a user to take an artifact and provide a reference to the registry components of the Semantic Infrastructure framework, and provide basic annotations.
Arrow: Other Platforms Integration	This integration represents the interaction of Semantic Infrastructure services with applications and platforms that might need to utilize function of the Semantic Infrastructure.
Box Name: SI Application Toolkit	This Semantic Infrastructure tool provides libraries and functions that ease the creation of new Semantic Infrastructure Framework enabled applications.
Box Name: Forms and Object Modeler	This Semantic Infrastructure tool is used to create forms models, message models and other core object models from defined structures. This tool works with information in the Semantic Infrastructure to access meta-models and model definitions to construct representations of objects which can be used for data collection and information exchange.
Box Name: Artifact Publication	This Semantic Infrastructure is the non-portal version of the artifact publication found in the Semantic Infrastructure portal. This component is different, because it will provide greater access to various components, enhanced governance support and manipulation of Knowledge Store objects requiring enhanced behaviors.
Box Name: SI Enabled Applications	This represents any number of applications that might need access to Semantic Infrastructure functionality and would utilize the Semantic Infrastructure application toolkit. This may include NCI applications such as caTissue.

6.2 - Overview of Semantic Infrastructure 2.0 Capabilities and Services

6.2 - Overview of Semantic Infrastructure 2.0 Capabilities and Services

February 23, 2011 Working Draft

These Semantic Infrastructure 2.0 Capabilities and Services Profile pages include the following. These pages are described below.

- [0 - Core Semantic Infrastructure Capabilities and Services Profile To Requirement Map](#)
- [1 - Artifact Management](#)
- [2 - caGRID 2.0 Platform and Terminology Integration](#)
- [3 - Clinical Data Forms Definition and Modeling](#)
- [4 - Conformance Testing](#)
- [5 - Service Discovery and Governance](#)

Semantic Infrastructure 2.0 Capabilities and Services Profile

This document describes all the central Semantic Infrastructure 2.0 capabilities and services, organized by functional profiles that address the following requirements:

- [Artifact Management](#)
- [Terminology and Platform Integration](#)
- [Forms Modeling](#)
- [Conformance Testing](#)
- [Service Discovery and Governance](#)
- [Reasoning and Inference](#)

The functional decomposition of Semantic Infrastructure 2.0 reflects the structure of the requirements. Each functional profile is a grouping of capabilities corresponding to identified Semantic Infrastructure requirements. In addition all requirements identified during Gap Analysis (which immediately preceded roadmap development) have been merged into relevant functional profiles as capabilities with appropriate traceability to the Gap Analysis requirements.

The Semantic Infrastructure is a Semantic Service Oriented Architecture that conforms with the principles and guidelines specified by the corresponding OASIS Reference Models, Ontologies, and Architectures. Within this document, these standards are represented as Semantic Profiles. Conformance with these standards has architectural implications for the business-oriented functional profiles. These architectural implications are reflected as extended capabilities of each functional profile, with traceability to the appropriate Semantic Profile conceptual model for Service-Oriented Architecture (SOA). This traceability, and the associated contextual description, help provide a bridge between the domain-specific terminology used to express requirements and the terminology used to describe the Semantic Infrastructure architecture.

The functional profiles are organized by the **Semantic Infrastructure Requirements** as follows:

- **1 - [Artifact Management](#)** The Semantic Infrastructure supports management, analysis, annotation, publication, query, and transformation of artifacts, including static and dynamic models.
 - **1.1 - [Administer Artifacts](#)** Artifact lifecycle management defines profiles to manage the lifecycle, governance, provenance, versioning, and representation of artifacts, as well as the relationships between artifacts.
 - **1.2 - [Analyze Artifacts](#)** Analyze Artifacts defines profiles supporting the analysis of artifacts utilizing semantic queries, reasoning, and rules.
 - **1.3 - [Model and Annotate](#)** Model and Annotate defines profiles supporting models, including model maintenance, constraints, bindings, extensions, and semantic annotations.
 - **1.4 - [Publish](#)** The ability to publish and discover information models will be supported by the semantic infrastructure, and the platform will leverage these capabilities.
 - **1.5 - [Search and Access](#)** The Semantic Infrastructure enables integrated access, search, and visualization of artifacts using a variety of search criteria, model serialization formats, and user interaction empowerment tools.
 - **1.6 - [Transform](#)** Transform defines profiles for management and application of transformations to support multiple views, serialization formats, inter-operability, semantic convergence, model migration, model merge and compare, and provisioning of target artifacts.
- **2 - [caGRID 2.0 Platform and Terminology Integration](#)** The Semantic Infrastructure supports seamless integration with the caGRID 2.0 platform.
 - **2.1 - [Export](#)** Event Processing and Notifications enables monitoring of services in the ecosystem and provides for asynchronous updates by services, effectively allowing a loose coordination of services that both provide and respond to conditions (possibly defined in business rules).
 - **2.2 - [Search and Access](#)** This group of capabilities focuses on enabling developers of composite services and applications to discover, compose, and invoke services. This includes the discovery of published services based on service metadata and the generation of client APIs in multiple languages to provide cross-platform access to existing services.
- **3 - [Case Report Forms Definition and Modeling](#)** Case Report Forms are the primary channel for capturing information in the healthcare and clinical domain. Forms also play a key role in information exchange and are critical to supporting interoperability in healthcare.
 - **3.1 - [Administer Forms](#)**
 - **3.2 - [Create Forms](#)** Tools and services for defining form templates.
 - **3.3 - [Search and Access Forms](#)** - Access models and reusable segments to support Case Report Form Definitions (not yet posted).
- **4 - [Conformance Testing](#)** Services specifications developed by NCI and the community have to be testable to ensure that the implementation conforms to the specification.
 - **4.1 - [Create Conformance Statements](#)**
 - **4.2 - [Search and Access Conformance Statements](#)** Conformance testing leverages the artifact and service metadata to validate that an implementation adequately addresses the requirements stated in the service specification. An example of service requirement is the ability to specify a response time in the specification (design time) and validate that this response time is valid for an implementation of the service. Additional test points include but are not limited to binding to specific terminologies and domain models.
 - **4.3 - [Test for Conformance](#)** Testing for SOA combines the typical challenges of software testing and certification with the additional needs of accommodating the distributed nature of the resources, the greater access of a more unbounded consumer population, and the desired flexibility to create new solutions from existing components over which the solution developer has little if any control. The purpose of testing is to demonstrate a required level of reliability, correctness, and effectiveness that enable prospective consumers to have adequate confidence in using a service.
- **5 - [Service Discovery and Governance](#)** Service discovery and governance allows service developers to specify rich metadata about services. This enables better discovery, and governance of services.
 - **5.1 - [Administer Services](#)** Administer Services defines profiles for management of service metadata and service classification schemes.
 - **5.2 - [Analyze Services](#)** Analyze Services defines profiles for service analysis, providing support for determining characteristics including service interaction dependencies, service reuse, service conformance assessment, heterogeneous data interchange, and service collaboration compatability.
 - **5.3 - [Search and Access Services](#)** Search and Access Services defines profiles supporting the discovery and visualization of

- services.
- [5.4 - Service Governance and workflows](#) This includes predefined templates, workflows, and governance policies for governing the service lifecycle as well as an approval and review process for service specifications and the ability to promote services through the stages of the service lifecycle.
- [5.5 - Service Policies](#) Service policies help establish constraints on the service specifications and mandate an approach. Policies can be specified around governance, access control and other design and runtime constraints.
- 6 - Reasoning and Inference: Refer to [NCI Enterprise Services Inventory Blueprint](#), Utility section, box U18 "Semantic Decision Support."

The semantic infrastructure capabilities and services address requirements and use cases for each domain. In addition to the domain specific use-cases, the capabilities also address CBIIT internal development and architecture requirements. Specifically, CBIIT has standardized on Service-Oriented Architecture as the foundational principle for applications architecture and interoperability. CBIIT has also adopted a formal approach (Enterprise Conformance and Compliance Framework) for defining service specifications. The capabilities address both the requirements for supporting semantic interoperability, and the need to publish formal specifications that can be adopted by external organizations and vendors.

The search and access profiles for each requirement are different as they are customized to access specific information in a defined format. For example, platform tools like the orchestration engine will leverage specific metadata to determine compatibility between two services.

6.3 - Tools for Semantic Infrastructure 2.0

6.3 - Tools for Semantic Infrastructure 2.0

February 23, 2011 Working Draft

This section provides an overview of the key tools that are expected to address the Semantic Infrastructure requirements. Tools are a combination of applications, user interfaces and services that address a specific Semantic Infrastructure 2.0 requirement. The tool description explains how users may work with the tools to perform specific tasks.

This information below describes the interactions a user would have with the Semantic Infrastructure components. These activities address aspects of the Enterprise Conformance and Compliance Framework (ECCF), including artifact registry and functional interactions.

This section includes the following:

- [Semantic Infrastructure Services and Interactions](#)
 - [Artifact Search and Access](#)
 - [Artifact Authoring](#)
 - [Artifact Governance and Lifecycle Management](#)
 - [Artifact Analysis](#)
 - [Access Control and Policy Management](#)
 - [Search and Access Services](#)
 - [Administer Services and Specifications](#)
 - [Analyze Services](#)
 - [Service Governance and Workflows](#)
- [Plug-ins, Loaders and Miscellaneous Development Tools](#)
- [Forms Editor](#)
- [Conformance Testing Framework](#)
- [Knowledge Based Systems Support Tools](#)

Semantic Infrastructure Services and Interactions

Artifact Search and Access

- Search the registry for model
- Utilizing a Common Terminology Services 2 (CTS 2) implementation, search for existing value sets
- Search the Identifiers registry for an appropriate identifier definition
- Retrieve an existing model from the registry
- Retrieve schemas and form definitions
- Import a new model for annotation (binding to terminology and value-sets)
- Retrieve the value set from CTS 2 implementation and apply to the model attributes

Artifact Authoring

- Create a new value set (leveraging terminology services) and manage existing artifacts
- Register the object identifier with its metadata
- Register a new artifact in the registry with its metadata (including ECCF specific information) and traceability to support provenance
- Link the models to existing models, domains, or both and classify artifacts

- Constrain existing models
- Constrain existing datatypes
- Link artifacts to service specifications
- Support collaborative development of metadata

Artifact Governance and Lifecycle Management

- Initiate artifact workflow
- Monitor the workflow process
- Allow curators to set the status of a model and receive change notifications
- Customize review and approval workflows
- Govern collaborative development

Artifact Analysis

- Execute semantic queries for search and analysis

Access Control and Policy Management

- Establish templates and policies that can be applied to artifacts
- Link the users security and access rights to particular governance and workflow
- Execute semantic queries for analysis
- Identify dependencies

Search and Access Services

- Discover services
- Execute complex queries to identify services
- Discover contracts at runtime

Administer Services and Specifications

- Create a new service specification
- Create and edit service metadata
- Link the artifacts to the service specification
- Construct conformance statements
- Run validations on specification and supporting artifacts
- Apply constraints, rules and policies
- Constrain and reuse existing specifications and artifacts

Analyze Services

- Execute semantic queries for analysis
- Perform impact analysis to identify dependencies and constraints

Service Governance and Workflows

- Initiate service specification governance
- Monitor the workflow and approval process
- Allow governance team to set the status of a specification and promote
- Manage versioning and localization of the service specifications
- Govern collaborative development

Plug-ins, Loaders and Miscellaneous Development Tools

Plug-ins and development tools are extensions to existing tools to support better integration with the artifact registry:

- Publish artifacts from eclipse and other modeling tools to the artifact registry
- Publish content from external tools to the artifact registry, with support for [topic maps](http://topicmaps.com) and DITA ([presentation on topicmaps.com](http://presentation.on.topicmaps.com))
- Extract and load metadata from artifacts into registry

Forms Editor

The Forms Editor is an end user tool for creating and editing forms:

- Create form definitions
- Retrieve models, datatypes and value sets
- Define skip patterns, layout and display options
- Publish forms to the artifact registry
- Reuse and leverage predefined templates

Conformance Testing Framework

The Conformance Testing Framework is a suite of tools for developing automated conformance testing that can be executed against an artifact or service:

- Retrieve or infer artifact metadata and any conformance statements provided
- Structure the artifact metadata for automated analysis
- Evaluate the artifact metadata and conformance statements against the base semantic profile
- Execute conformance tests
- Generate conformance report

Knowledge Based Systems Support Tools

Knowledge based systems support tools are knowledge engineering tools to support reasoning and inference use cases:

- Knowledge Representation Definition
- Rule Pattern Definition
- Domain Model Development
- Knowledge Base Model Definition
- Fact Generation and Mapping tooling
- Rule Generation tooling
- Reasoner Framework for Standard Knowledge-based System Development
- Automated Reasoning Service Publication

6.4 - Tie-in with Terminology and Platform

6.4 - Tie-in with Terminology and Platform

February 23, 2011 Working Draft

This section describes the operational dependencies between the semantic infrastructure and terminology and the platform and includes the following:

- [Dependencies between Semantic Infrastructure 2.0 and caGrid 2.0](#)
- [Semantic Infrastructure Overview](#)
- [Semantic Infrastructure Registry](#)
- [Registry Reliance on Platform](#)
- [Platform Reliance on Registry](#)
- [Metamodel and Information Model](#)

Dependencies between Semantic Infrastructure 2.0 and caGrid 2.0

Refer to the following sections of [6 - Dependencies Between Semantic Infrastructure 2.0 and caGrid](#) in the caGrid 2.0 Roadmap Documents

- Semantic Infrastructure Registry
 - Registry Reliance on Platform
 - Platform Reliance on Registry
- Metamodel and Information Model

Semantic Infrastructure Overview

In an effort similar to developing this roadmap for Semantic Infrastructure 2.0, a team is developing a roadmap for the future platform, security and tools, caGrid 2.0. The Semantic Infrastructure 2.0 will be tightly integrated with the runtime caGrid 2.0. The purpose is to achieve a more comprehensive approach to computable semantic interoperability than is possible with the existing integration between caDSR and caGrid 1.x. With the adoption of SAIF (Service-Aware Interoperability Framework) and Enterprise Conformance and Compliance Framework (ECCF) and the introduction of behavioral semantics, the infrastructure of the grid must provide increasingly sophisticated support to leverage and enforce behavioral specifications.

The notion of "computable semantic interoperability" (CSI) applies semantics to not only the static data passed between machines, but also to the behavioral and functional operations exposed for coordination of behaviors during the interaction. Likewise, the semantic infrastructure itself (that is, its tools and applications) are being transformed to fully participate in the new services-aware environment. Thus the Semantic Infrastructure will depend on the grid platform as at least one of potentially many delivery platforms for its information.

The Semantic Infrastructure 2.0 is expected to provide management services and tooling comparable to those which exist today (including vocabulary services, model management and annotation, and curation tools and others as needed), albeit in potentially new formats and standards. However, it also has the increased scope of greater flexibility in accommodating granular levels of conformance and participant sophistication.

For example, the assumption of adherence to a centrally curated authoritative source of information models and terminology is no longer true; the

infrastructure must gracefully accommodate local terminologies or localizations as well as standard terminologies. It must enable a path towards "as much interoperability as is possible" between any two parties, rather than enforcing full "compatibility" of all participants. Additionally, the Semantic Infrastructure 2.0 is charged with making the wealth of knowledge contained within the numerous SAIF artifacts available and consumable (in a programmatic fashion) to all the grid participants. Having runtime support for leveraging this information to inform and drive service interactions is a key value proposition for the future platform and semantic infrastructure.

Semantic Infrastructure Registry

The key components of the Semantic Infrastructure 2.0 are still being identified and scoped. The Semantic Infrastructure Registry has been identified as a key component. It is expected to be critical to the grid. The Semantic Infrastructure Registry may ultimately be manifested as numerous types of registries and services; potentially there may be numerous instances of each.



Note

The ECCF registry provides storage for the semantic components of ECCF artifacts as specified by governance. The ECCF Registry will be populated by the iterative process of service design and specification.

The specification of a CBIIT enterprise service specification requires the development of three separate artifacts:

- The CIM (computationally independent model specification)
- The PIM (the platform independent model specification)
- The PSM (the platforms specific model) specifications

The CIM, PIM and PSM are again a collection of artifacts (models). The ECCF matrix is placeholder for these artifacts organized by Reference Model of Open Distributed Processing (RM-ODP) viewpoints and model-drive architecture (MDA) perspectives.

Currently a Microsoft Word document acts as a template or placeholder for describing the Computation-Independent Mode (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM) (along with the artifacts of each viewpoint), while the future semantic infrastructure will define computable representation formats for this information.

Part of this computable representation is expected to be a Service-Oriented Architecture (SOA) ontology for describing various entities involved in the numerous conformance assertions (with examples including but not limited to services, operations, data types, faults, and actors). This ontology will provide the backbone for reasoning to be performed by the platform and tools at both runtime and design time (as illustrated later in this section).

Registry Reliance on Platform

While the Semantic Infrastructure Registry services will be specified in ECCF, and potentially manifested on multiple platforms, one such platform will be the grid and will therefore use the platform as scoped in this document.

The registry will require numerous capabilities described in this document including the security layer for items such as authentication, authorization, auditing, and data assertions and integrity. The infrastructure may also require support for a rules engine capable of consuming and enforcing rules in both static (for example, data constraints in the information model) and behavioral semantics (for example, pre-conditions and post-conditions of operation invocation) stored in the registry.

Similarly, the platform will inform the content and format of various platform specific artifacts to be stored in the registry (including but not limited to XML Schema (XSD) and Web Service Definition Language (WSDL)). The platform will provide the capability to enforce or test conformance to those profiles by, for example, checking service interfaces against published PSMs and doing data validation against published information models. Finally, the platform will act as the service implementation technology for the grid services of the registry (that is, be the management interfaces or consumer facing services).

Platform Reliance on Registry

The platform itself will require and leverage numerous capabilities of the Semantic Infrastructure 2.0, most importantly, access to the information contained in the ECCF registry. The registry will facilitate nearly all parts of the service development and consumption life cycle.

At design time, the registry will provide a wealth of information to the designer including available relevant service specifications to adopt and extend, information models and terminologies to leverage for new operations, and formal specifications of expected behavior of the existing services that the new service may consume from in its implementation. For example, service templates (shelled out implementation artifacts) could automatically be constructed based on platform specific specifications.

Extending beyond the basic query and retrieval of these artifacts, tools can be built to actually understand the semantics of this information and aid the service developer. For example, potentially relevant information models may be found by entering simple terms like "tissue sample" into a tool, which binds that string terms to concepts in identified terminologies and locates models containing information bound to those terms. Similarly, behavioral contracts may be discovered based on terminology binding to their function based on simple search terms like "data insert," which can act as models or examples for new service operations.

Further, such "understanding" of behavioral and static semantics can provide a powerful feature in a tool for workflow or service composition. It could leverage this information to make suggestions on "next steps" in a workflow, even suggesting specific services to use. It could also provide powerful integrity checking of the data flow and functional effect, by validating that the invocations are consistent with published conformance assertions and rules.

At deployment time, the platform (or deployment tools built on it) can automate the generation and execution of a test suite to check conformance assertions published in the registry, relevant to the service being deployed.

At run time, the service can provide powerful self-descriptive metadata by referencing profiles, policies, conformance assertions, and specifications in the ECCF registry. This metadata will provide significant details about the nature and behavior of the service, and can be used to discover it, as well as to ascertain programmatically how to correctly consume it (and validate it is functioning correctly). The platform may also be able to automatically flag non-conforming service instances (for example, services sending incorrect data, or running outside of published performance metrics) by monitoring runtime behavior.

Metamodel and Information Model

The Semantic Infrastructure 2.0 effort is still deciding on the format and structure to be used. This decision will be important to caGrid 2.0, as it will inform how things like the publishing of service metadata work, and how higher layer semantics (for example, operation preconditions) are built upon static descriptions (for example, WSDL). It is expected, however, that a transition from ISO 11179 metadata to RIM-derived semantics is important in the future infrastructure. As further information is available from both roadmap efforts, this section will evolve to discuss the impact on the platform.

7 - Gap Assessment for Semantic Infrastructure

7 - Gap Assessment for Semantic Infrastructure

February 23, 2011 Working Draft

The section provides an assessment of the gap between the roadmap and existing tools and platform. The following topics are included:

- [Existing NCI Semantic Infrastructure](#)
- [Proposed Features in Semantic Infrastructure 2.0](#)

Existing NCI Semantic Infrastructure

The NCI semantic infrastructure currently consists of a suite of tools aimed at terminology curation of models submitted as UML XMI files for semi-automated annotation; terminology services for concept lookup and codesystem browsing; and basic terminology and ontological relationships in the NCI Thesaurus and Metathesaurus. This bundle of infrastructure applications together with model-driven software engineering tools are termed caCORE (Cancer Common Ontologic Representation Environment).

caCORE tools and APIs are developed by the National Cancer Institute Center for Bioinformatics and Information Technology (NCI CBIT) to provide the building blocks for development of interoperable information management systems. This suite of tools has helped to enable interoperability and data sharing from the scientific bench to the clinical bedside and back with the current semantic infrastructure.

caCORE includes the following key components:

- [EVS](#) (Enterprise Vocabulary Services) for hosting and managing vocabulary
- [caDSR](#) (Cancer Data Standards Registry and Repository) for hosting and managing metadata
- [caCORE SDK](#), the GUI-based [caCORE Workbench](#), and associated tools for model-driven software engineering of systems which can be easily integrated with caGrid.

[EVS](#) and the [caDSR database and tools](#) are the current basis of the semantic foundation for interoperable data and analytical services at NCI. caDSR is based on the ISO 11179 Part 3 metadata standard.

Developers use caCORE components to create "caCORE-like" systems. By definition these systems have object-oriented information models registered in caDSR whose meaning is linked to EVS vocabularies, and have open, public APIs and web services to provide access to the data. The [caBIO data service](#) is an example of a caCORE-like system developed using caCORE components.

Using caCORE tools, developers adapt and build applications that are [caBIG® compatible](#), that is, interoperable with other caBIG® tools.

caCORE tools include the following:

- [caDSR APIs Download](#)
- [CDE Browser; DTDs](#)
- [Form Builder](#)
- [CDE Curation Tool](#)
- [caDSR Administration Tool](#)
- [UML Model Browser](#)
- [Semantic Integration Workbench](#)
- [caDSR Sentinel Tool](#)
- [NCIThesaurus](#)
- [NCIMetathesaurus](#)

Additionally caCORE includes the [caCORE workbench](#), a tool with a graphical user interface (GUI) to facilitate the creation of a caBIG® silver or gold compliant system. The caCORE Workbench acts as a process guide and an integrated platform, enabling the user to more readily create a Data or Analytical service on the Grid. The following caBIG® process workflows are supported:

- Creation of a UML Model (ArgoUML, Enterprise Architect)
- Semantic integration (SIW, CDE Browser, UML Model Browser, Curation Tool)
- Model mapping (caAdapter)
- Application creation and deployment (SDK)
- Creation of a grid service (Introduce)

Proposed Features in Semantic Infrastructure 2.0

Semantic Infrastructure 2.0 is meant to provide a means of fully supporting the existing NCI semantic infrastructure, while providing a means for ongoing transformation of the existing artifacts and creation of equivalent tooling to support all current functionality of the semantic infrastructure.

Semantic Infrastructure 2.0 extends the current functionality of the semantic infrastructure by adding the following functionality:

- A new means of assessing conformance of artifacts and applications to improve software development and semantic consistency
- A semantically linked artifact repository for easy discovery of the registry contents
- A metadata repository that links to the artifact repository
- A cross-artifacts editing dashboard that allows model artifacts to be linked to other artifacts such as terminology value sets
- A rules engine for operating on the artifact repository and metadata repository to enable dynamic annotation and the comparison of artifacts
- A reasoning platform that executes inferencing and links to rule engines enabling the discovery of implicit information rather than explicit information only
- Introduction of additional semantic modeling standards (ISO 21090, HL7 Reference Information Model (RIM), Semantic Web Languages (Web Ontology Language (OWL), Resource Description Framework (RDF)) in order to handle the broad requirements of enabling simpler query functions and enriched data discovery
- A more automated artifact governance platform that includes the ability for community input to governance decisions
- Multiple model transformation tools and APIs
- Tools for authoring standards-compliant artifacts including schemas, models, and terminology value sets
- Tools for authoring forms using the new semantic models in order to meet the demands of customers who require these so that they can meet meaningful use requirements, and who want full semantics for data aggregation and discovery
- Broad use of Model Driven Architecture technologies
- Close integration with caGRID 2.0

The table below shows a high level view of the gaps between what the current semantic infrastructure provides and what Semantic Infrastructure 2.0 will provide for several use case-driven functionalities.

Requirement	Current Semantic Infrastructure	Semantic Infrastructure 2.0	Gap closed
Retrieve any artifact	CDE	Domain models, Logical models, terminology, documents, forms, behavioral models an specifications	Ability to retrieve any artifact in context
Manage artifacts	CDE Curator only	Open to all	Ability for anyone to annotate an artifact and submit to governance
Service discovery	Constrained to service discovery on caGrid	Service discovery tied to artifacts that can link to data provision	Ability to discover a service, its links to other services, the service contract, the artifacts that are behind the service
Bench to Bedside Form creation	Clinical research form creation	Form creation of any healthcare, clinical research, or life science form	Supports all form users and conforms to Office of the National Coordinator (ONC) requirements for meaningful use forms
Decision support across artifacts	None	Semantic linkage across multiple artifacts, inference of implicit knowledge about the artifacts and their relations	Provides enhanced search and retrieval of artifacts and extends the metadata for any artifact through inference of relations
Conformance Testing	None	Semantic reasoning and inference with automated classification, relations, and traceability of artifacts	Provides the full traceability and conformance testing for artifacts in a standard framework (Enterprise Conformance and Compliance Framework (ECCF))
Data discovery	Able to query caDSR for a model attribute and return an attribute identifier and reuse that identifier in a query for data	Semantic inference, semantic to relational adapters and scalable relation graphs relate services to artifacts, artifacts to terminology, and terminology to data allowing queries of models, classes, concepts or any other artifact and its data	Provides the ability to link services to each other and to the explicit definitions of the data they provide

8 - Migration Strategy and Ongoing Support for Existing Customers

8 - Migration Strategy and Ongoing Support for Existing Customers

February 23, 2011 Working Draft

This section will describe the Semantic Infrastructure 2.0 Roadmap Team's recommendation for a migration strategy from current infrastructure to the new infrastructure. The Semantic Infrastructure team in concert with other CBIIT teams is working on prototyping processes and tools to support migration. The working plan is that the existing infrastructure will be used until the new infrastructure is available. CBIIT believes there will be a one time migration of existing content from existing infrastructure (that is, the caDSR) to the new infrastructure. The new infrastructure is planned to support existing users of the current infrastructure, including seamless transition for existing applications calling the data service API of the current infrastructure. The prototyping effort through March 2011 will test these assumptions and arrive at a more concrete proposal based on the migration prototyping work.

9 - CBIIT Project Recommendations

9 - CBIIT Project Recommendations

February 23, 2011 Working Draft

- Assessment Report
- Tools Recommended for Semantic Infrastructure Work
 - Triple Store Access
 - Semantic Knowledge Store
 - Data Conversion and Artifact Access
 - Integration Support
 - Inference, Rules and Expert Systems
 - RDF and OWL Tools
 - General Purpose
 - Flow Management for Services, Processes and web Applications
 - Design
 - Component Repository

Assessment Report

This section will provide the Assessment Report generated during the prototyping work of the inception phase.

Tools Recommended for Semantic Infrastructure Work



Note

Supported CBIIT application structures, such as the Java platform, Tomcat, JBOSS, Ant, and Maven, are discussed in the architecture sections of this roadmap and the caGrid 2.0 Roadmap, [6 - caGrid 2.0 Architecture](#) and [6 - Semantic Infrastructure 2.0 Architecture](#). Additional standards are discussed in section [10.4 - CBIIT Project Recommendations](#) of the caGrid 2.0 Roadmap and on the [VCDE WS Standards Efforts](#) page.

The tools and libraries listed here have been identified as being of possible help in certain aspects of the Semantic Infrastructure 2.0 Roadmap project and development of the architecture. The tools listed below have not been formally described as supported by CBIIT at this time. However, they suggest the type of components and architecture expected to best satisfy project requirements.

Triple Store Access

Access tools represent methods for interacting with information represented using Resource Description Framework (RDF) and Web Ontology Language (OWL) representations of metadata. In addition, these tools may provide additional support for inference engines.

Jena (sourceforge.net) - The Jena Semantic Web Framework is a tool for building semantic web applications. It provides a programmatic environment for RDF, RDF Schema (RDFS) and OWL as well as a SPARQL engine and rule-based inference. This is a general purpose tool that supports combinations of models in memory as well as transactional persistence of triples. Jena works well with ontologies with large numbers of classes and individuals which require read and write functionality.

OwlAPI (sourceforge.net) - The OWL API is a Java API for reference implementation for creating, manipulating and serializing OWL ontologies. This interface provides a fast, in-memory representation for manipulating OWL ontologies and provides persistence to OWL in XML format.

Sesame (opendrf.org) - Sesame is an open source RDF framework with support for RDF Schema inferencing and querying.

ARQ (openjena.org) - Supports the W3C standard for RDF queries. This is packaged with Jena and can be used against Jena models.



Note

These systems can be used in combinations. For example, OWL files created with OwlAPI can be read by Jena. However, Jena TDB is specific to Jena, and you must use Jena to access it. See below for persistence methods.

Semantic Knowledge Store

Stores represent the ability to store RDF and OWL representations. These tools provide various features including support for larger datasets, transactional updates, and integration with traditional relational databases. Although the tools listed below are open source, there are also non-open source tools which provide similar functionality including AlegraGraph and Oracle 11G. These system may also use Jena or Sesame for access.

Jena TDB (openjena.org) - This is an integration between the Jena model factory which provides a high speed persistence of RDF triples. Writes to this system are not fully transactional, and so care must be taken to manage transactions externally, where required.

Jena SDB (openjena.org) - Provides a persistence model using a variety of relational database management system (RDBMS) back ends. Utilizing JDBC connections, SDB persists into traditional tables the RDF triple information. This solution is fully transactional, but suffers from insert and query limitations for performance.

Sesame SAILS (opendrf.org) - Sesame provides a way to integrate various representations for access (called SAILS (Storage And Inference Layer). Sesame SAILS have been created for in-memory, relational database, and a variety of other formats.

OwLIM (ontotext.com) - OwLIM is implemented as a Sesame SAIL and comes in both an open source and commercial implementation. OwLIM boasts that it is the most scaleable semantic repository in the world (the commercial implementation). It also offers a high speed reasoning system built into the system.

db2rq (wiwiss.fu-berline.de) - Not technically a general purpose store, db2rq provides a semantic reference layer to an existing RDBMS environment, allowing for SPARQL and other interactions within a given environment.



Note

These systems can also interact. Specifically Jena can access Sesame through a model factory component and Sesame can access Jena through a SAIL. This does not indicate compatibility, but rather an abstraction. This means that the database tables created by Jena SDB do not match the RDBMS tables generated by Sesame. However, as an example, code written using Jena as an interface, if written correctly, can be independent of the persistence method.

Data Conversion and Artifact Access

These are tools which aid in the programmatic access to artifact sources to aid in transformations and processing.

poi (apache.org) - Poi is a general purpose tool for accessing Microsoft documents in the .DOC, .XLS, and traditional Microsoft proprietary formats. In addition Poi supports the openXML standard including Microsoft .DOCX and related formats through the openxml4j api. Information about this standard can be found [on openxmlpl.biz](http://openxmlpl.biz).

OBO-Edit (oboedit.org) - The editor provides an API for accessing OBO-based ontologies.

Eclipse EMF (eclipse.org) - The ecore component of Eclipse EMF is a tool for accessing models in XMI for conversion to other representations. Other aspects of the Eclipse EMF may also be useful.

In addition there are standard access tools for accessing RDF and OWL representations. See [Semantic Knowledge Store](#) above.

Integration Support

Spring (springsource.org) - Spring provides a number of components which are designed to either ease the adoption of new technologies, or to provide greater control over certain integrations. The number of Spring components is large; some significant components include Spring Framework, Spring Flow, Spring Web Services, and Spring Security. These components are all based on certain core patterns that make the components more flexible.

Inference, Rules and Expert Systems

These are defined as a way to provide methods of representing and executing decision support, orchestration, analysis and many other aspects of application functionality. They share a way to represent certain behaviors for which a more concise language has been created than traditional programming languages. Some inference and rule systems support standards such as OWL DL (Description Logics), or RuleML (Markup

Language), or RIF (Rule Interchange Format). In addition, there may be extensions or additional functionality which make them suitable.

RDF and OWL Tools

Pellet (clarkparsia.com) - Pellet is an OWL 2 (partial) reasoner providing the core classification functionality. Pellet is broadly used and integrated into various platforms including Protege 4 and TopBraid Composer. Pellet is written directly in Java and so can easily be integrated into other java applications directly without external configurations or implementations.

Fact++ (owl.man.ac.uk) - Fact++ is an implementation of an OWL 2 (partial) reasoner written in C++. Fact++ requires the implementation of a component which is accessed by applications. Because it is written in C++ it has the potential to be faster than Java implementations.

Hermit (hermit-reasoner.com) - Hermit is an OWL 2 reasoner implementing a high performance algorithm in Java. It is dependent on the OwlAPI.

TopBraid SPIN (topquadrant.com) - TopBraid SPIN is an implementation of the SPARQL Inferencing Notation. It is an open source implementation, and can be integrated in a number of ways. It has many uses including an RDF constraint language, a rules language, a SPARQL function language (used as a way to extend SPARQL), and a method of storing reusable queries. Envisioned and implemented by TopBraid, it expands functional behaviors in ways that are impossible to declare in DL, or where it would be inappropriate. Use of TopBraid SPIN requires the use of the Jena API.

General Purpose

Jess (jessrules.com) - Jess stands for Java Expert System Shell. Jess is an implementation of the rete algorithm and supports a number of rule definition languages. It is the reference implementation of JSR 94 standard for java rule engines. Jess supports CLIPS (C Language Integrated Production System) and RuleML languages, as well as its own XML representation of CLIPS. Jess provides many ways to extend the functionality into Java Applications in both direction (able to call java functions from the rules, as well as call rule functionality from java). Jess is available without cost for academic uses as well as through various commercial licenses. It does not have a cost for development, as there is a trial download that times out after a number of days, and can be re-downloaded.

Drools Expert (jboss.org) - Drools is a component of the JBoss community. Drools is described as a business logic integration platform. It has a number of components which may be integrated to provide different support including a managed rule repository. Drools is an implementation of the rete algorithm. Drools supports a proprietary language as well as an XML representation of its own language. Transformations of RuleML to Drools may be available.

Flow Management for Services, Processes and web Applications

Open ODE (apache.org) - Open ODE is an Apache project which utilizes the Web Services Business Process Execution Language (WS-BPEL) standard for organization of work flow. It is supported by Apache ServiceMix and can be used to manage web service choreography where that is appropriate.

Drools Flow (jboss.org) - Drools flow is an integration of the Drools rule engine designed to manage business or process flow. Drools Flow definitions can be rendered in the Business Process Modeling Notation (BPMN) notations, but an Eclipse plugin is also provided for visual design of workflows. Drools Flow also works with [Drools Guvnor](#) to provide a repository of workflows, and provides audit and control over workflow processes. Drools Flow has built in support to provide monitoring of flow activities. Drools Flow is an implementation of Business Process Modeling (BPM).

Bonita Open Solution (bonitasoft.com) - The Bonita Open Solution is an implementation of BPM that provides an environment for designing, managing and executing flow control. Flows in Bonita are designed graphically and can be executed directly through deployment as applications, or uploaded to a functional engine for execution. Bonita provides both an API for integration, and a web-based tool to provide execution points for state transitions and management.

Spring Flow (springsource.org) - Using an integration with the Spring MVC (model-view controller), Spring Flow allows for the definition of flows which control activity within a given session. This tool separates the page flow from the business logic, allowing for many alternative flows using the same pages. This simplifies applications which perform activities in different modes (create versus edit) or through different means (create "Wizards").

Design

OWL and RDF provide the ability to represent information as metadata and as functional components of a system. As a result, individuals may produce RDF or OWL ontologies which will be integrated into the fabric of the system. In addition to the standard tools supported by CBIIT relating to design, the use of good ontology editors will help promote the consistency of representation and functionality. In some senses these are integrated development environments (IDEs) in the fact that development occurs; however, they can also be considered as Platform-specific Model (PSM) design tools because the output becomes a documentable model representation.

Protégé 4 (stanford.edu) - Protégé is developed and supported at Stanford University and has been used for ontology development for many years. Protégé 4 is an attempt to reach beyond the frame-based roots of Protégé and provide a newly envisioned representation of OWL ontologies. Protégé 4 utilizes the OwlAPI for accessing OWL ontologies and so shares the limitations of the OwlAPI. Specifically Protégé 4 does not support persistence to general purpose triple stores, and must be able to load the ontology entirely in memory. However, the use of the OwlAPI in design and editing where it is appropriate, gives Protégé 4 a performance advantage in the loading of ontologies, and provides unique functionality as related to ontology integration. Protégé 4 is an Eclipse Rich Client Platform (RCP) Application.

TopBraid Composer (topquadrant.com) - TopBraid Composer is available in both a community edition and a commercial license version.

TopBraid Composer Community Edition provides an Eclipse plugin-based approach to ontology editing. This allows for the integration of other tools via the OSGI standard, and provides a basis for using other Eclipse-based tools. TopBraid Composer uses the Jena tool to access ontologies and shares its limitations. In addition, the community edition is limited to the editing of OWL or RDF files and does not support access of database stores. It does support SPARQL as part of its functionality, and utilizes the Eclipse approach of projects.



Note

Since there is no open source or community tool for accessing stores other than text files, many developers use Protege 4 or TopBraid Composer, and then create scripted or programmed solutions to upload models into those stores. However, there are other significant additions to functionality in the commercial versions of TopBraid Composer that are not addressed here.

Component Repository

There are some current repository systems that may help in the management of elements such as rules and flow controls.

Drools Guvnor (jboss.org) - Drools Guvnor is a tool which provides access to a common rule repository, flow repository and other aspects of the Drools system, providing browsing and access control. In addition it integrates with graphical editor for rules and flows.

10 - Semantic Infrastructure 2.0 Interim Development

10 - Semantic Infrastructure 2.0 Interim Development

February 23, 2011 Working Draft

This section will provide information about interim development for Semantic Infrastructure 2.0, based on results from the [Migration Prototyping project](#).

The migration strategy for Semantic Infrastructure includes:

- Support for current users of CDEs
- Transforming a set of CDEs to the new Semantic Infrastructure representation
 - Determining the process for the transformation
 - Determining how many CDEs need to be transformed, including usage and re-use
 - Determining whether parts of process can be automated
 - Determining the level of effort
 - Determining what the user experience will be, that is, what interfaces community members will use

For more information about prototyping for the Semantic Infrastructure 2.0 and caGrid 2.0 Roadmaps, refer to [CBIIT Roadmaps Inception Phase and Prototyping](#).

11. Semantic Infrastructure 2.0 Roadmap References and Glossary

11. Semantic Infrastructure 2.0 Roadmap References and Glossary

February 23, 2011 Working Draft

The [Glossary for CBIIT SAIF, caGrid 2.0 and Semantic Infrastructure 2.0](#) on the CBIIT SAIF Wiki provides lists of acronyms and definitions of key terms.

A list of [caGrid 2.0 and Semantic Infrastructure 2.0 - References](#) is provided.